ANALYSIS OF FLOW SEPARATION OVER AERODYNAMIC AIRFOILS

Sergio Jesús Rodríguez Sánchez

October, 2014

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

BACHELOR THESIS

Aerospace Engineering

Tutor: Óscar Flores Arias

Contents

List of Figures

List of Tables

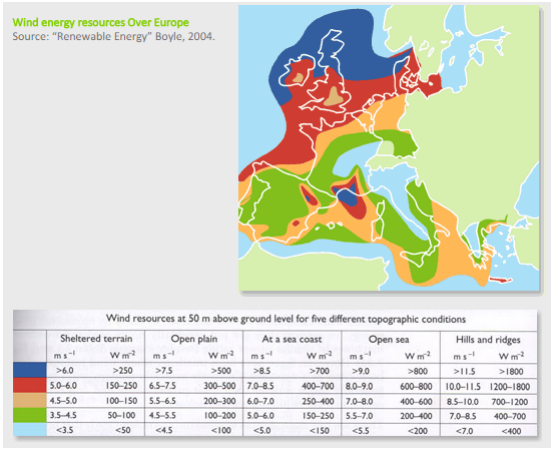
1 Introduction

1.1 Motivation

Energy industries are nowadays struggling with the most efficient source of energy generation. The list of energy resources is huge starting with traditional coal and going through the different technologies until reaching the controversial nuclear reactors. One of the most interesting areas in energy generation is the renewable energy technologies. Among the different renewable energy sources, this project will be focused on eolic energy generation via horizontal axis wind turbines (HAWTs). The major concern for turbine manufacturers is how to increase the efficiency of the turbines they produce in order to obtain the maximum amount of electric energy out of the available wind sources. That is the reason why manufacturers invest enormous amounts of money on improving the technology that supports their business. Improving the aerodynamic efficiency of turbine blades is one of the most representative investments that manufacturers are performing beside improvements on material science and wind forecasting. Specifying the scope of this project, the three main areas that will be considered are detailed hereinafter:

1. Analysis of eolic energy technologies and its state of the art.
2. Analysis of mathematical equations governing the physical phenomena regarding air flows over eolic turbine blades.
3. Improvement of eolic turbine blades airfoil profiles.

The analysis and study of these new energy sources is a field that offers plenty of opportunities and a deeper study and further improvement of both, the mathematical model and devices that are used to study the production of these new energies, is needed. For the purposes of this project, focus and efforts will be placed on the study of eolic energy. The rationale supporting that choice is due to the fact that Spain is a country where the suitable land conditions could be found and, as it is shown on Figure 1, Spain also has a suitable wind resource with not extreme winds but strong enough to provide eolic power without breaking eolic facilities.



Wind resources at 50 m above ground level for five different topographic conditions

	Sheltered terrain		Open plain		At a sea coast		Open sea		Hills and ridges	
	m s^{-1}	W m^{-2}	m s^{-1}	W m^{-2}	m s^{-1}	W m^{-2}	m s^{-1}	W m^{-2}	m s^{-1}	W m^{-2}
	>6.0	>250	>7.5	>500	>8.5	>700	>9.0	>800	>11.5	>1800
	5.0–6.0	150–250	6.5–7.5	300–500	7.0–8.5	400–700	8.0–9.0	600–800	10.0–11.5	1200–1800
	4.5–5.0	100–150	5.5–6.5	200–300	6.0–7.0	250–400	7.0–8.0	400–600	8.5–10.0	700–1200
	3.5–4.5	50–100	4.5–5.5	100–200	5.0–6.0	150–250	5.5–7.0	200–400	7.0–8.5	400–700
	<3.5	<50	<4.5	<100	<5.0	<150	<5.5	<200	<7.0	<400

Figure 1: Eolic resources in the European Region

Data for Figure 1 was obtained from the table shown under the chart..

It is a fact that wind turbines among other renewable energy sources such as solar and geothermal, are called to replace the use of traditional power plants based on combustion of solid fuels (oil, coal and Natural) gas. Nowadays, according to [3] the generation of electric power as of August 30th, 2014 is shown in Figure 2.

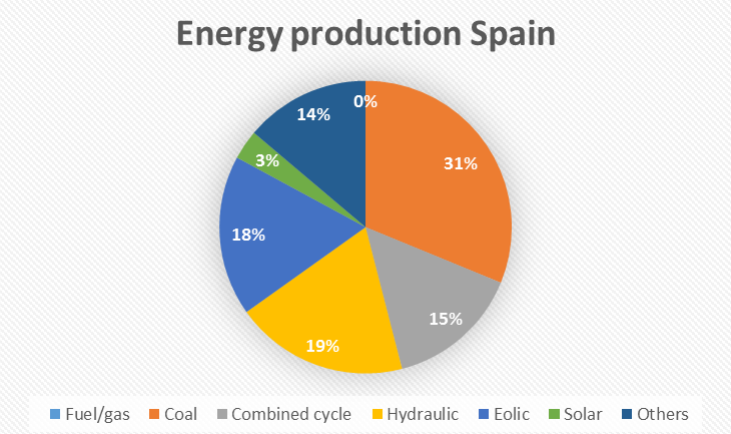


Figure 2: Electric Power Generation. Aug. 30^{th}, 2014.

As it can be appreciated in this info-graphic recreated with data from Electric Spanish Network (REE in Spanish), renewable energies supply 40% of Spanish electric power. This numbers need to increase because, according to the latest predictions based on Hubbert's oil peak prediction[1], petroleum and oil reserves are decreasing, and after 2020, the continuous decrease of oil production rates will become critical. According to projections of the U.S. Energy Information Administration, energy consumption in all the regions of the planet is not doing anything but growing. The average growth for the World-wide estimations is +1.5% of energy consumptions for the period between 2010 and 2040. Data supporting this statement can be found on the official EIA website [5]. Therefore, it is important to move and evolve, using new sources of renewable energy and leaving behind the traditional energy sources whose reserves are becoming scarce.

On the market, a variety of products that allow the analysis of physical phenomena related to energy production could be found. Some of the capabilities these software are able to provide are: heat transfer and semiconductor materials analysis for solar energy study purposes, air flow analysis for wind power estimation and aerodynamic improvements, water flow analysis for hydroelectric power, and the list is as long as any application that could be imagined. The fact is that the tools and software used to study these technologies have some licenses associated whose cost represents a relevant research and development investment for engineering companies.

Concerning the scope of this project, eolic energy and air flow analysis, FLUENT is the most widely used commercial software. For more information about Ansys Fluent, see [6]. But there is a whole world besides ANSYS software: companies devoted to aerodynamic analysis and wind-related applications have developed their own Computational Fluid Dynamics (CFD hereinafter) software for their individual purposes. All these software are private programs that cannot be accessed and even less modified, because of that, open-source software has become the best alternative

[1] Hubbert's peak theory states that considering a particular geographical area and for a given individual region, the rate of petroleum production follows a bell-shaped curve. Early in the curve, production rates increase due to discovery and exploitation; after the peak, production rates decrease due to resource depletion. It is based on the currently known fuel reserves and the assumption of exponential growth and decrease according to the region and reserves constraints. Ref. [4]

for engineers who want to develop their own projects trusting on a reliable program that is able to fulfill their needs and expectations. Open-source software has the advantage that there is no license fee associated with its use: the code is free for everyone and you can edit it and adapt it to your requirements as long as it stays free for all the community. On the other hand, it has also a disadvantage: since there is not an authority regulating that software, software reliability is based on users knowledge. If the code created by a certain user is based on wrong assumptions or has any bugs, the community may find errors, or the results of the performed computations may not be accurate. For the purpose of this project, we will use and compare the results obtained with open-source software named OpenFoam (see more info in [7]) and the analysis tool XFLR5 (see more info in [8]). Further information regarding these software will be provided in the following sections.

For the readers who have never been introduced to open-source software, according to the official OpenFoam website: "The OpenFOAM® CFD Toolbox is a free, open source CFD software package which has a large user base across most areas of engineering and science, from both commercial and academic organizations. OpenFOAM has an extensive range of features to solve anything from complex fluid flows involving chemical reactions, turbulence and heat transfer, to solid dynamics and electromagnetics. It includes tools for meshing, notably snappyHexMesh, a parallelised mesher for complex CAD geometries, and for pre- and post-processing. By being open, OpenFoam offers users complete freedom to customize and extend its existing functionality, either by themselves or through support from OpenCFD. It follows a highly modular code design in which collections of functionality (e.g. numerical methods, meshing, physical models, ...) are compiled into their own shared library." Case set-up and instructions on how to run OpenFoam to reproduce the cases studied in this project are shown in further sections of this project.

In order to understand the physical phenomena in which these software rely, the definition of wind turbines operation, airflow over airfoils, boundary layer and turbulence are required and will be covered in the upcoming sections of this project.

1.2 Wind turbines

1.2.1 Eolic Energy

Eolic energy is one of the most widely used sources of renewable energies as it is shown in Figure 2. It is defined as the power that the wind can produce by moving the blades of a wind turbine. That motion moves the main shaft inside the turbine rotor and the generator transforms that mechanical movement into electric energy. The basis of eolic energy are out of the scope of this project but for further study on that field, the basic knowledge about this energy source are provided in [9] as an overview of such a wide topic. One of the advantages of eolic energy is that during energy generation it produces no CO_2 emissions and does not contribute to the greenhouse effect. On the other hand, manufacturing of wind turbine's blades and towers does produce pollution. Improvement on "green" manufacturing processes that reduce the environmental impact of this energy source is another subject for study but is out of the scope of this project. Apart from this, wind energy creates many jobs in many different fields: engineering, manufacturing, material development, transportation, installation, maintenance... That is a crucial issue if we consider the current socio-economic scenario.

Wind energy exhibits also some drawbacks: the first and most easily appreciated is the visual impact that wind farms produce on landscapes. People leaving nearby these wind farms are sometimes against the installation because these big structures modify the nature and the habitat of animal species. The second one is the noise that wind turbines produce. This noise comes from both mechanical (rotor) and aerodynamic (blades) sources. Noise impact reduction has become a critical issue for wind turbine manufacturers because it represents a competitive advantage when a new wind farm is planned. This factor becomes of extreme severity in areas where populations are within a few kilometers from the wind turbines. Policies such as revenue sharing with the towns that accept wind farms within their territories improve the acceptance of this technology. For countries where there is not plenty of firm land, offshore wind farms are a suitable solution in order to obtain the advantages that wind energy provides.

Wind speed is the most relevant factor for energy production. As it was shown in Figure 1, wind sources distribution is not homogeneous for all territories. Hence, accurate weather forecasting is required on the early stages of wind farm placement analysis. As it was explained in a previous paragraph, wind turbines transform the kinetic energy on the wind to mechanical energy due to blades rotation and shaft rotation and, in the last stage and using a generator, they obtain electrical energy out of the rotational mechanical energy.

According to [10], the maximum available energy (P_{max}) is obtained by:

$$P_{max} = \frac{1}{2} m V_0^2 = \frac{1}{2} \rho A V_0^3 \tag{1}$$

Where m is the mass flow, V_0 is the wind speed, ρ is the air density and A the area where the wind speed is reduced by blades rotation.

Betz's limit [2]establishes that there exists a maximum power coefficient (C_P)

$$C_p = \frac{P_{real}}{P_{max}} \tag{2}$$

assuming that the actual power obtained is lower than the maximum available power. This maximum value for the power coefficient is $Cp_{max} = 16/27 = 0.593$. Most of modern turbines operate in the neighborhood of this limit and are optimized but there is still room for improvement in the airfoil design and that is one of the goals of this paper.

For windy countries, the price of electricity [$\$/kWh$] obtained from eolic energy is competitive in comparison with the prices of conventional energy production such as coal or fuel power plants. In order to decrease the production cost of this energy, turbine manufacturers are focusing on reducing the cost of the materials used for the turbine, blades and tower, and also improving the efficiency of both the generator and the aerodynamic profiles that compose the turbine blades. There is an important and relevant regulation for our country concerning wind energy. These regulations include interest rates, cost of land where wind farms are installed and other factors such as bonuses for the use of renewable energies can be found on the last government BOE publication "Law 24/2013, December 26th, regarding the electricity industry" [12].

1.2.2 Wind turbines review

Even though airfoil profiles have been used for aircraft since the beginning of 20^{th} century when National Advisory Committee for Aeronautics (NACA) started developing and researching about the optimization of airfoil profiles for aircraft, airfoils used on the first modern wind turbines were not optimized for the environments were turbine blades usually work, conditions with higher angles of attack and twist angles.

NACA 6 series have been widely used at the early stages of the wind turbine development era, due to the knowledge that engineers had about these airfoils. Nowadays, manufacturers are using their own optimized airfoil profiles for wind turbines. Engineers are developing their own CFD tools for improvement and optimization of those designs. Regarding other difficult conditions that turbine blades face during their performance, different materials with improved strength and stiffness have been developed. Most of them use similar materials with different composition than the ones used for aircraft: glass fiber and carbon fiber. Improved manufacturability is required due to the fact that turbine blades present higher twist angles than the ones that could be appreciated on civil and military aircraft. In Figure 3, the differences mentioned before between aircraft and wind turbine blades are manifested.

[2]Betz's law calculates the maximum power that can be extracted from the wind, independent of the design of a wind turbine in open flow. The assumptions taken for Betz's development are: 1. The rotor does not possess a hub, this is an ideal rotor, with an infinite number of blades which have no drag. Any resulting drag would only lower this idealized value. 2. The flow into and out of the rotor is axial. This is a control volume analysis, and to construct a solution the control volume must contain all flow going in and out. 3. The flow is incompressible. Density remains constant, and there is no heat transfer. Further literature on this topic can be found on Ref. [11].

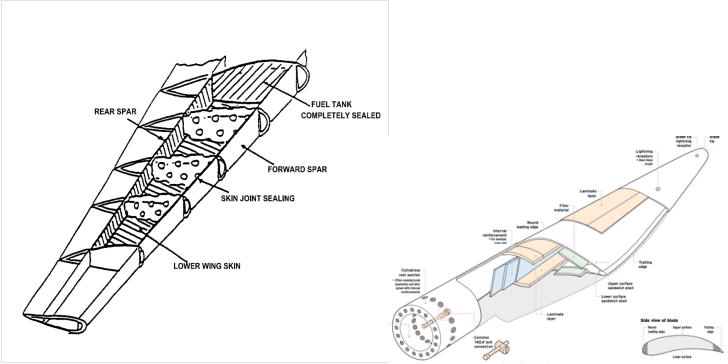


Figure 3: Wind turbine blade vs. Aircraft wing.
Images for the comparison are obtained from [1] and [2].

During the development of wind turbines, several wind machine configurations have been established according to [13].

1. Drag type turbines
2. Lift type turbines (two types: vertical or horizontal axes)
3. Magnus effect turbines
4. Vortex turbines

A brief description of each type will be presented hereinafter in order to present the scenario and establish the rationale for the choice of the turbine type that will be studied. According to the data presented on previous sections, the most reasonable choice is to choose horizontal axis lift type turbines because, among other reasons, they present the most affordable manufacturing processes since they are very similar to airplane wing production.

Essentially, most of modern wind turbines are of the lift type, and over 90% of these are of the horizontal axis type according to [13]. Magnus effect and vortex plants are only developed for academic or very particular purposes but they are not commonly used for energy production. Modern wind turbines consist of a number of rotating blades looking like propeller blades but with bigger sizes. According to the most usual classification, when blades are connected to a vertical shaft, the turbine is called vertical-axis wind turbine, VAWT, and, if the shaft is placed horizontally, the turbine is called a horizontal-axis wind turbine, HAWT. Wind turbine manufacturers have put all their efforts in building HAWTs, and this project will focus on that particular type of wind turbine. The main parameters defining the aspect of a HAWT are: rotor diameter, number of blades and tower height.

An example of this type of turbine is shown in Figure 4. This picture was obtained from [10].

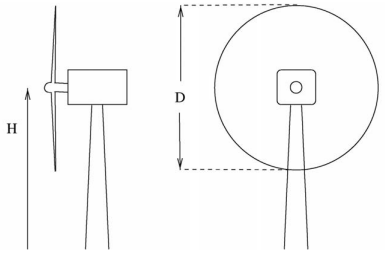


Figure 4: Horizontal Axis Wind Turbine

In the following paragraphs, a brief overview of the different types of wind turbines will be provided.

- Drag-Type Wind Turbines: in this type of wind turbine, the wind exerts a force in the direction of the air stream. The air pushes on the surface and generates the mechanical motion. There is a physical condition that clearly states that the surface on which the wind is exerting the force cannot move faster than the wind itself. Persian civilization used this type of machines. In the next figure, an example of how this machine worked obtained from [10] is shown.

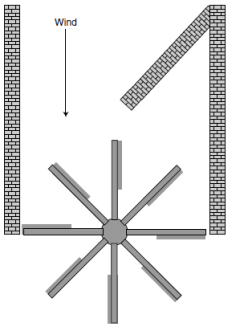


Figure 5: Drag-Type (Persian) Wind Turbine

- Lift-Type Wind Turbines [14]: in this type of wind turbines, the wind exerts a perpendicular force to the direction of its main flow stream. In this case, the blade of the turbine can rotate faster than the wind if the angle of the blade is the appropriate one. Figure 6 shows this type of devices. The blades are hollow and they have a hole on the tip in order to push out the air inside them taking advantage of the centrifugal action.

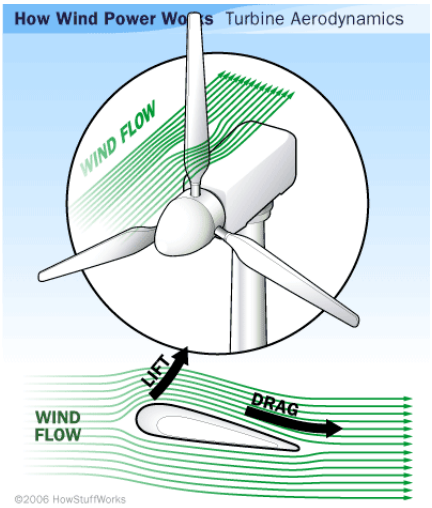


Figure 6: Lift-Type Wind Turbine

- Magnus turbine effect [15] is the one responsible of, among other things, the "curve" that balls can achieve in games like in baseball. The example according to [13] uses is the one of a pitcher throwing a curve: "the pitcher causes the ball to spin, creating an asymmetry: one side of the ball moves faster with respect to the air than the other and, consequently, generates the "lift" that modifies the trajectory of the ball. An identical effect occurs when a vertical spinning cylinder is exposed to the wind." The resulting force is the one used to move the wind turbine blades.

Figure 7: Magnus Wind Turbine

- Vortex Wind Turbines [16]: the previous turbines (Lift and Magnus turbines) where based on the normal direction of the wind, but it is also possible to extract energy from the wind by making it enter tangentially into our device. As a consequence of that fact, the air inside is gyrating and the resulting centrifugal force causes a radial pressure gradient to appear. The spinning air exits through the exhaust of our turbine, forming a vortex continuously swept away by the wind.

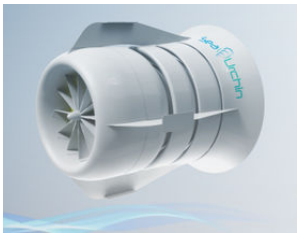


Figure 8: Vortex Wind Turbine

1.2.3 Wind turbines parameters of study

Equations governing the full operation of a wind turbine are a complex set of equations starting with the Navier-Stoke's equations that rule the behavior of the air flow over the turbine blades and finishing with the production trends that rule the energy generation that a given wind turbine will provide to the electric grid. A small review on the main concepts that rule the performance of wind turbines has been provided in this project, but in this section, further definition and development of the most relevant parameters for a wind turbine will be shown.

As it was mentioned in Section 1.2.1, the power that a given turbine can provide is the most important parameter of study for wind turbines. That is because the equation that sets the power

that a wind turbine can provide, relates the energy source (wind) with the reason eolic turbines are made for (electricity production).

- Turbine power output (P_T): maximum power that a given turbine can provide to the electric grid considering wind speed and total turbine efficiency (accounting for mechanical and electrical losses):

$$P_T = \frac{1}{2}\rho A V^3 C_p \eta [W] \tag{3}$$

  Where: ρ is the air density, A is the swept area of the turbine blades, V is the wind speed, $C_p$is the power coefficient according to the definition provided in Section 1.2.1($C_{p_{max}} = 16/27 = 0.593$) and η is the turbine efficiency ($\eta < 1$). According to the conditions for C_p and the fact that there are always mechanical and electrical losses, the best a wind turbine can do in terms of power delivered is always smaller than 59% of total power available according to wind speed. The optimization of the wind turbine blades airfoil, increases the value of C_p making it as close to 16/27 as possible. And that is the goal of this paper, by delaying the separation point of the blade and making it close to the trailing edge, the value of power coefficient is therefore increased.

- Blade aspect ratio AR: it is defined for any generic surface that faces an airstream. It is the ratio between the length (S symbol will be used because it is most commonly named "span" for wings) and the mean aerodynamic chord ($\bar{c} = \frac{Surface}{Span}$) of the surface.

$$AR = \frac{S}{\bar{c}} \tag{4}$$

  In Figure 9, a visual description of blades aspect ratio is shown.

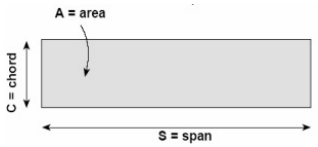


Figure 9: Wing/blade Aspect Ratio

- Number of blades: since every blade disturbs the airflow, it is important to minimize the number of blades as much as possible always without compromising the efficiency of the full mechanism. Because of structural requirements, at least two blades are required for moment balances. More than 4 blades make the turbine heavy and non-cost effective. The compromise between cost, structural reliability and aerodynamic efficiency provide that the most reasonable choice is to build turbines with 3 blades because it is cost affordable and provides a reasonable amount of flow disturbance with a good overall efficiency.

1.3 Air flow over 2D airfoils

The motion of any fluid is governed by Navier-Stoke's equations. These equations in general form are expressed below where ρ is the density of the flow, v is the velocity vector, p is the pressure distribution, τ is the stress tensor of order two and f the body forces acting on the fluid.

$$\rho(\frac{\partial \mathrm{v}}{\partial \mathrm{t}} + v \cdot \nabla \mathrm{v}) = -\nabla \mathrm{p} + \nabla \cdot \tau + \mathrm{f} \tag{5}$$

In the particular case of 2D airfoils, and using a coordinate system placed on the surface of the airfoil, only two components of the velocity are considered: normal (y-axis) and tangential velocities (x-axis). The velocity component on the spanwise direction (z-axis) is considered to be equal to zero. The effect of the this velocity generates a distribution of pressure on upper and lower surfaces of the airfoil that cause vertical and horizontal forces on the airfoil profile. The reacting force F from the flow is decomposed into a direction perpendicular to the velocity at infinity V_∞ and to a direction parallel to V_∞ . The former component is known as lift (L) and the latter is called drag (D) (see Figure 10). Lift and drag coefficients c_l and c_d are defined as: $c_l = \frac{L}{\frac{1}{2}\rho V_\infty^2 c}$ and: $c_d = \frac{D}{\frac{1}{2}\rho V_\infty^2 c}$; where ρ is the air density and c the chord of the airfoil profile and V_∞ is the free stream wind velocity.

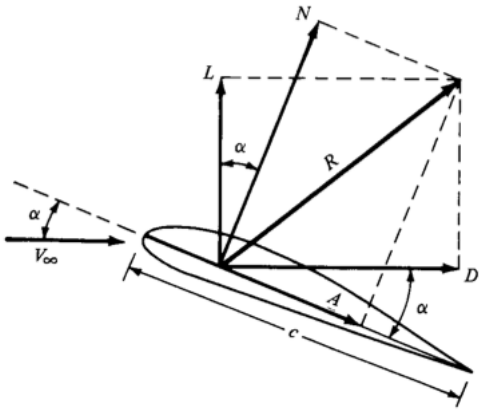


Figure 10: Lift and Drag forces

The coefficients c_l, c_d and c_m are functions of α, Re and Ma.

- α is the angle of attack defined as the angle between the chord-line and V
- Re is the Reynolds number based on the chord (c), the flow speed V, $Re = \frac{cV}{\nu}$, where ν is the kinematic viscosity of the flow
- Ma denotes the Mach number, it is defined as the ratio between the flow speed (V) and the speed of sound (a) $Ma = \frac{V}{a}$.

For a wind turbine, drag and moment coefficients are only functions of α and Re. For a given airfoil the evolutions of c_l, c_d and c_m are shown in polars. An example of the polar for the S809 airfoil computed with XFLR5 is shown in Figure 11.

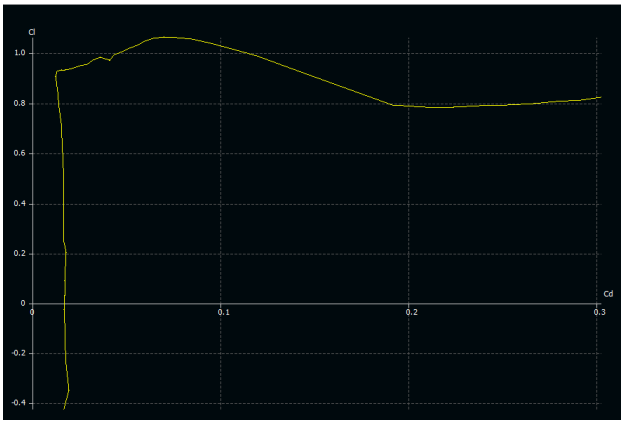


Figure 11: S809 polar

To completely describe the forces, it is also necessary to know the moment M about a point in the airfoil. This point is often located on the chord line at c/4. For conventional purposes, this moment is positive when it tends to pitch the airfoil nose up and the moment coefficient is defined as:$c_m = \frac{M}{\frac{1}{2}\rho V_\infty^2 c^2}$. In our field of study, pitching control is critical for wind turbine blades. Pitching control allows turbines for optimizing the angle of attack and to avoid structural damages due to high speed rotations or high wind speeds.

From basic fluid mechanics it is known that a pressure gradient, $\partial p/\partial r = \rho V^2/r$, is necessary for the airflow to be curved and go through the airfoil profile. Far away from the airfoil profile, the atmospheric pressure must be lower than atmospheric pressure on the upper side of the airfoil and higher than atmospheric pressure on the lower side of the airfoil. This pressure difference between the upper and lower surfaces of the airfoil profile is responsible of generating the lift force.

It is important to introduce the concept of boundary layer even though it will be studied with more detail in Section 1.4 of this project. Boundary layer is defined as the region of the air flow stream that is in direct contact with the airfoil physical surface. When the airfoil has zero or low α, boundary layer remains attached to the surface and drag is minimized and caused by friction with the air (viscosity terms of Navier-Stoke's equations). In this situations we are in the laminar flow regimes.

With a deeper understanding of the polars for a given airfoil, it can be appreciated that c_l increases linearly with α, until a certain value of α, where a maximum value of c_l is reached. After that point, the airfoil is said to stall. Stall occurs when boundary layer detaches from the airfoil surface generating turbulence that breaks the pressure difference between upper and lower surfaces of the airfoil profile. For small α, drag coefficient c_d is almost constant, but increases rapidly after stall. The explanation for this phenomena again relies in the way the boundary layer detaches from the upper surface of the airfoil and the turbulence that it generates once it detaches the surface. We can find different scenarios according to the separation position of the boundary layer:

- When separation starts at the trailing edge: soft stall because pressure peak is not disturbed (pressure peak is located near the leading edge of the airfoil and most of the lift is generated on this region of the airfoil).
- On the other hand, if separation starts at the leading edge of the airfoil, the entire boundary layer may separate almost simultaneously with a dramatic loss of lift.

The behavior of the viscous boundary layer is very complex and an attempt to describe this phenomena will be accomplished on the next section.

1.4 Boundary Layer

1.4.1 Boundary layer description

If we want the solution of Navier-Stoke's equations to represent a large Reynolds number flow, we need to consider the existence of surfaces where velocity suffers a discontinuity, i.e. where velocity goes from a value different from zero at some point of the flow to zero at a different point (in this case, the surface of the aerodynamic profile), due to the effect of viscous terms. The region where the speed decreases from this value different from zero up to zero is called boundary layer. Within this region, viscosity has a considerable effect and causes velocity to vanish at the surface of the profile to be analyzed.

In Figure 12, the streamlines that comprise the boundary layer region are shown as well as the leading edge stagnation point caused by the adverse pressure gradient the the flow finds when it approaches the surface of the airfoil.

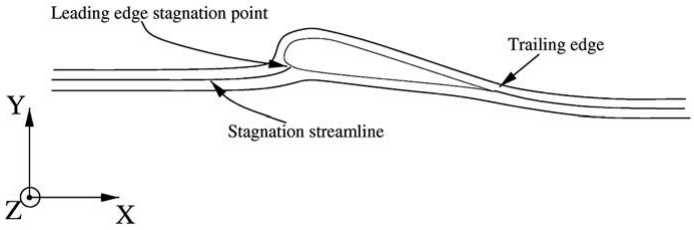


Figure 12: Flow over 2D airfoil

As it was mentioned on the previous section, stall is closely related to separation of the boundary layer. The main goal for blade manufacturers is, therefore, to reduce drag and avoid separation. By delaying separation, or in other words, keeping boundary layer attached to the wind surface, drag is reduced and aerodynamic efficiency is increased. That is the main goal of this project: improve the aerodynamic efficiency of the selected profile and estimate the distance that the boundary layer separation was delayed.

Boundary layer size is important for the purposes of separation point estimation. The thickness of the boundary layer in this project will be defined where the velocity in the x-direction $v_x = 0.99v_e$. This boundary layer thickness is shown in Figure 13

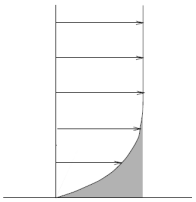


Figure 13: Boundary layer thickness

Boundary layer is an upper surface problem because the fluid flowing over the airfoil accelerates as it passes the leading edge. On the lower side of the airfoil, the curvature of the airfoil is smaller and the flow decelerates and separation on lower surfaces of airfoils is an strange phenomenon. At the trailing edge, pressures from the upper and lower surfaces must be the same in order to satisfy the Kutta condition.

Laminar airfoils are those on which the boundary layer is laminar and attached in the most part of the upper surface of the airfoil surface. This airfoils do present separation and turbulence but for higher values of α. For small values of α, the flow accelerates and the pressure gradient is negative on the upper surface avoiding separation. At some point, the flow needs to be decelerated in order to respect Kutta condition and at the point where the flow starts to decelerate is a suitable point where the boundary layer is likely to detach. Just after this transition point, boundary layer thickness is still small. Due to the continuous deceleration near the trailing edge, the boundary layer cannot withstand the positive pressure gradient and separation is more likely to occur. It is important to control the point where separation will occur because once the flow becomes turbulent it is hard to control and to predict the behavior of that flow on the remaining part of the airfoil's surface.

1.4.2 Equations governing boundary layer behavior

According to [17], the equations governing the velocity field of the boundary layer are:

- Continuity equation:

$$\partial v_x/\partial x + \partial v_y/\partial y = 0 \tag{6}$$

- Momentum equation:

$$\rho\partial v_x/\partial t + \rho v_x \partial v_x/\partial x + \rho v_y \partial v_x/\partial y = -\partial p_e/\partial x + \mu\partial^2 v_x/\partial y^2 \tag{7}$$

- Together with the boundary conditions that include the nonslip condition and the value of the external velocity outside the boundary layer:

Nonslip condition: $y = 0 : v_x = v_y = 0$
External velocity: $y \to \infty (y > \delta) : v_x = v_e(x,t)$

1.4.3 Separation point

For non-ideal flows and bodies with curvature, boundary layer separates at some point of the surface of interest. At the separation point, a vortex sheet is generated and potential flow solutions are no longer valid. The condition that determines the location of the separation point is determined by the "zero-shear stress" condition $\partial v_x/\partial y = 0$, and at that point, the momentum equation is reduced to $\partial p_e/\partial x = \mu\partial^2 v_x/\partial y^2$; and, as it was mentioned before, separation will only occur at points with positive pressure gradients.

1.4.4 Relevance of separation in airfoil optimization

As it has been mentioned a couple of times during this project, delaying boundary later separation for a turbine blade profile is crucial fact for turbine manufacturers. By delaying separation, pressure drag coefficient (c_{dp}) is reduced, lift coefficient evolution is improved delaying stall. Increasing lift coefficient has an impact on the losses of the turbine and it also affects the aerodynamic efficiency of the turbine affecting as a consequence the maximum power that the turbine can provide. Structural integrity of the blades and the rotor needs to be ensured and, since pitching mechanisms of the blades are very accurate and sensitive parts, if a big gust or continuous turbulence are stressing the mechanism, malfunctioning of the system will be an inevitable consequence. In the end, it is all related to make the flow go over the blade on a smoothly manner in order to avoid instabilities, or unexpected movements that may damage the structure. Smooth, laminar flows, are also more efficient for the aerodynamics of the turbine and that implies that more power can be achieved by getting closer and closer to the Betz's limit.

1.5 Tubulence

1.5.1 Turbulence review

Turbulence is defined by the irregular motion of fluid particles and the complete chaos derived from the interaction and fluctuations of the values of the velocity of those fluid particles. Examples of turbulent flows are shown in the following pictures: add nice pics of turbulent flows.

Turbulence peculiarities are the disorganized and "random" behavior of the fluid particles that comprise the flow. Turbulent flows seem chaotic and are very difficult to repeat: it is almost impossible to obtain the same turbulent flow twice because of the randomness on the motion of the particles. Turbulence is a 3D phenomena and the velocity of the particles has 3 components. That causes the flow to mix on all the possible manners and, therefore to be intermittent in time and space and almost impossible to predict. There is also a huge amount of dissipation of energy due to the interaction between the fluid particles.

Because of this list of characteristics; turbulence, even though it is also represented by Navier-Stoke´s equations (see Section 1.4), is a problem that has been frustrating engineers for years. It is still a difficult phenomena to understand, but CFD has improved the knowledge that fluid mechanics engineers can exploit. In order to simplify the solution of turbulent problems, Reynolds introduced the method of decomposing flow variables into a mean part (Average) and deviations. There are a few considerations that must be taken before studying turbulent flows:

1. The first one is that Navier Stoke's equations are non-linear since there are terms with different orders on their derivatives.

2. The first assumption that was made for 2D airfoils is to create an uniform flow in the spanwise direction. As a consequence of that, the mean velocity on that direction is zero. On the other hand, the fluctuations of the velocity are not zero on that direction but they are included in the turbulence model.

3. In case of using statistical formulations for turbulent models as Reynolds Averaged Navier-Stoke's (RANS), the system of equations to be solved presents less equations than unknowns. Therefore, solving problems with this method, creates what is called "the Closure problem". This problem appears because of the fact that with Navier-Stoke's and continuity, the set of equations to be solved adds up to 4 equations. The difficulty arises because these equations contain more unknowns. The unknowns are: 3 components of velocity and pressure (in the simplification for 2D airfoils there are only 2 components), and 6 components of the viscosity-stress second order tensor. This problem is associated with the non-linearity of turbulence. There is no exact solution to this problem because there are more unknowns that equations, but different approximations have been made by linearization of the non-linear terms as functions of the velocity components.

4. Due to the problems with the closure problem, Direct Numerical Simulations (DNS) seems to be the solution to the issues that appear when trying to solve turbulent flows, the problem with DNS is that it has an enormous computational time for the smaller scales and that implies an enormous amount of computational cost. This is the reason why approximation methods are employed for solving turbulent flows in the industry.

Nowadays engineers solve turbulent problems using one of these three principal methods:

- Reynolds Averaged Navier-Stoke's (RANS) Equations of Turbulence.
- Large Eddy Simulations (LES)

- Direct Numerical Simulations (DNS)

Each one of this methods has its own advantages and disadvantages that will be explained on Section 1.5.2.

1.5.2 Turbulence models brief description and selection criteria

- RANS: for this first turbulent model, the velocity components are split into a mean component and the deviations around it. Mathematically we have the expressions for pressure and velocity:

$$U(x,t) = U_{mean}(x) + u(x,t) \tag{8}$$

and

$$P(x,t) = P_{mean}(x) + p(x,t) \tag{9}$$

According to that, and using our Navier-Stoke's equations, the final form of the RANS Equations in 1D flow as the one for the boundary layer equations:

$$\frac{\partial(U_{x_{mean}} + u_x)}{\partial x} = 0 \tag{10}$$

for continuity equation and,

$$\frac{\partial U_{x_{mean}}}{\partial t} + U_{y_{mean}}\frac{\partial U_{x_{mean}}}{\partial y} = \nu\frac{\partial^2 U_{x_{mean}}}{\partial y \partial x} - \frac{1}{\rho}\frac{\partial P_{mean}}{\partial x} - \frac{\partial (u_x u_y)_{mean}}{\partial y} \tag{11}$$

for momentum equation.

The main drawbacks of RANS analysis are that it is never able to reproduce entire flow fields with accuracy because too much information is lost in the averaging process and more importantly, the turbulence model has constants that require to be tuned for every single application or problem to be solved.

- LES: turbulent flow is assumed to be composed of Eddies. These Eddies of size l have a velocity $u(l)$ and a time scale $\tau(l) = \frac{l}{u(l)}$. Reynolds number of this eddies is assumed to be large and, therefore, the effect of viscosity on dissipation can be neglected. They are unstable eddies and they transfer their energy to eddies of smaller size. These smaller eddies follow the same pattern and the energy cascade continues until viscosity is effective and dissipates the kinetic energy. For LES, the contribution of the big scales to momentum and energy equations is computed, but the effect of the smallest eddies is modeled. The way of solving a problem using LES is to decompose the flow variables, obtain a system of solved scale variables, model the viscous stresses for smaller scales and solve the equations with the selected model for the viscosity term. According to that process, the resulting continuity and momentum equations for our Navier-Stoke's set of equations become:

$$\frac{\partial U_{x_{mean}}}{\partial x} = 0 \tag{12}$$

for continuity equation and,

$$\frac{\partial U_{x_{mean}}}{\partial t} + \frac{\partial (U_x U_y)_{mean}}{\partial y} = \nu\frac{\partial^2 U_{x_{mean}}}{\partial y \partial x} - \frac{1}{\rho}\frac{\partial P_{mean}}{\partial x} - \frac{\partial \tau_{xy}}{\partial y} \tag{13}$$

for momentum equation. Where the term: $\frac{\partial \tau_{xy}}{\partial y}$ is called subgrid scale stress tensor and represents the effect of small scale eddies on the flow.

This method is more accurate than RANS analysis but there is still loss of information when modeling the smaller eddie scales. Since there are less constants to be tuned in this model, that makes this model more desirable.

- DNS: uses the capabilities of computers to solve the whole set of Navier-Stoke's equations with enough resolution to capture all the physical relevant scales. DNS is an expensive method because it consumes enormous amounts of processing capabilities and, therefore, its main application is to provide data for validation of LES models and discover fundamental concepts of turbulence. This method usually consumes large amounts of processors running because it solves for every size and time scales, providing billions of data points that need to be processed afterward. DNS is the most accurate treatment since it solves for every single point, size and time variable, there is no loss of information but it has a very high computational cost that increases more and more as Re increases. It requires also big amounts of physical resources as processors, memories and time.

It is required to reach a compromise between the accuracy of the results and the consumption of resources. The selection criteria for the most appropriate method needs to be based on the answer to the following questions:

1. What is the required level of description for the turbulent flow that is being analyzed?
2. Which are the costs and the required resources on time and processors?
3. Which is the range of applicability that the solution of this problem is going to achieve?
4. What is the expected accuracy of the results?

After making the required considerations, and answering the previous questions, the selected turbulent model for this project is Spallart Allmaras model for RANS. The rationale behind that choice is that by using OpenFoam, the tuning of the parameters mentioned before is affordable for the purposes of this project. The selection for an Spallart Allmaras modeling is because OpenFoam offers that capability without any need of coding or editing the turbulent model which will be a more advanced field of study.

1.5.3 Turbulent flows effect on wind turbines

Turbulence affects wind turbine performance in different ways:

The first effect is a poor aerodynamic performance. If there is separation and turbulence begins close to the leading edge of the airfoil, the values of c_d increase and that is translated into a poor aerodynamic efficiency. Most of the energy that wind can produce is lost in the wakes generated after the blades. The second effect is the structural damage that turbulence generates on the blades, gearbox, rotor and generator. Turbines are designed to withstand a maximum amount of loading, but if due to turbulence these loads increase, structural damages can be caused on the rotor or the control mechanisms inside the generator. When these loads are continuous on time, fatigue may affect the turbine blades and produce mechanical failure of the system. Turbulent gusts can also excite normal frequencies of the blades and cause damage to the structure, leading to economical losses and stoppage on electric energy supply. Finally, if blades are always moving in a turbulent flow, that flow can randomly change on speed or direction, causing decompensations on the loads that the blades are supporting. That causes structural damage and also can affect the twist of the blade and the pitching control mechanisms.

2 Objectives

2.1 Goals

The main goals of this project are listed below:

1. Redesign an airfoil in order to delay boundary layer separation and improve pressure distribution (c_p).
2. Validation of the new design using two open-source computational software: the first one is the aforementioned XFLR5 potential flow solver and the RANS solver OpenFoam.

All these goals will be achieved by running certain cases that will be explained in Section 4 of this project. The methodology and process will be explained on Section 3 and the optimization method used in order to improve pressure distribution will be explained in the following subsection.

2.2 Optimization method

Since one of the goals of this paper is to improve the pressure distribution on our selected profile, an optimization process will be performed in order to achieve the expected improved performance of our airfoil while validating the results obtained by the open-source software OpenFoam. This process comprises the following steps:

1. Profile selection and pre-processing for XFLR5 import and OpenFoam meshing.
2. XFLR5 Direct Analysis in order to obtain polars ($c_l$and c_d evolution) and pressure distribution (c_p) with estimation of boundary layer separation point.
3. After selecting a suitable AoA (α), run the case with OpenFoam and obtain pressure and velocity distributions for this profile.
4. XFLR5 Inverse Analysis to obtain the shape of an airfoil with improved pressure distribution.
5. Compare new shape with original airfoil in order to maintain the dimensions for similar manufacturing and repeat XFLR5 Direct Analysis for new airfoil.
6. Perform analysis of new airfoil with OpenFoam in order estimate separation point of the new airfoil.

With these steps, the results obtained by using an open-source will be compared with the ones obtained by using XFLR5 software. A comparison between the different pressure distributions and separation point estimations for the original and the new profile with the different software will be performed. In this project, two different scenarios were selected:

1. Scenario 1: Pressure drag coefficient(c_{d_p})improvement without affecting camber of the airfoil in order to delay separation.
2. Scenario 2: aerodynamic efficiency c_l/c_d improvement by changing camber and thickness distribution on the profile.

Both cases are analyzed using OpenFoam and the results for these simulations will be found on Section 5. There is also a required mesh validation process that is explained on Section 3.1.4.

3 Methodology

3.1 Profile

3.1.1 Selection Rationale

For the purposes of this analysis, it is important to select a suitable airfoil that is currently being used in the industry. In this case the selected profile was NREL's S809 airfoil that has been used for different applications during the past years. One of the main applications of this airfoil was its use for HAWTs blades. Since this whole project is focused on that certain type of wind turbines, the selection is appropriate and the profile will be studied because it can be used as a reference for further studies performed after this project.

3.1.2 Profile description

In order to minimize energy losses in wind turbines caused by the use of NACA non-optimum profiles, the National Renewable Energy Laboratory began to develop optimized airfoils for HAWTs. NREL's airfoils, are designed to have a $c_{l_{max}}$ which is non-sensitive to roughness effects according to NREL profiles report [18]. By making the transition from laminar to turbulent flow close to the trailing edge, the effect of surface roughness and its associated losses is minimized. Therefore, airfoils reduce turbulent boundary layer size, achieving lower values of drag and continuous laminar flow over most part of the airfoil.

In Figure14 the S809 airfoil shape is shown according to [18]

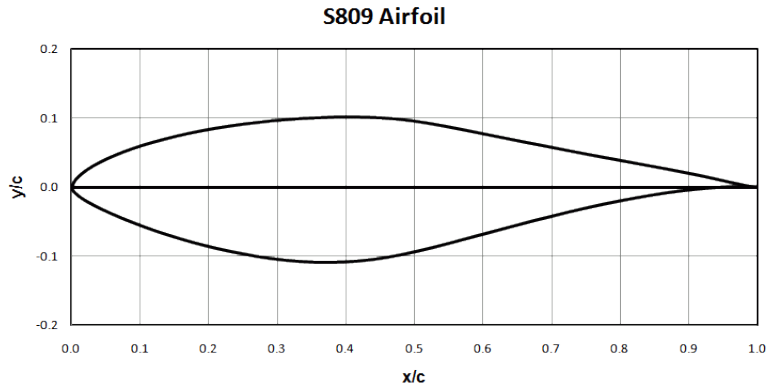


Figure 14: NREL's S809 airfoil profile

3.1.3 Profile mesh

In order to create the different meshes that are going to be used for the analysis and optimization of this profile, two different data sheets were created: one for XFLR5 data containing all the data points that define the airfoil profile and the other one is the mesh data file that would be used for OpemFoam simulations.

Data sheet for XFLR5 can be found on Appendix I and the following picture shows the profile to be used in this software.

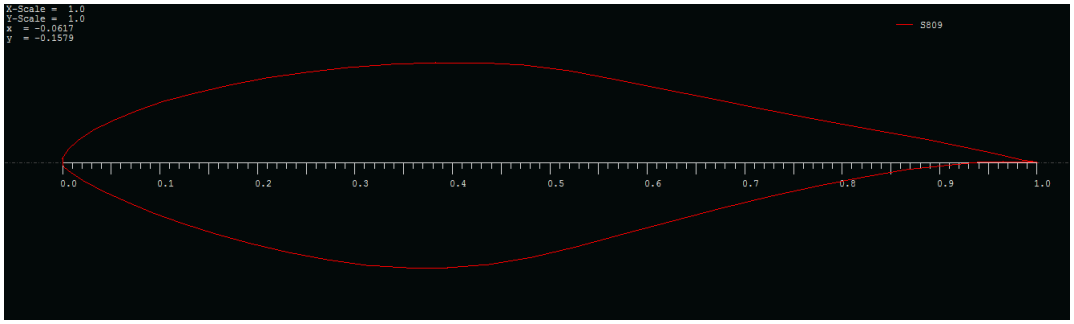


Figure 15: S809 airfoil in XFLR5

The mesh created for OpenFoam requires a deeper explanation because of the peculiarities of this software: airfoil data has to be provided beginning on the upper point of the trailing edge moving counter clockwise through airfoil surface until reaching the lower point of the trailing edge. In order to create a compatible file that can be run in OpenFoam, the data sheet that defines the mesh needs to fulfill certain requirements and to provide a certain structure. The structure corresponding to the mesh file required for OpenFoam simulation is shown in Section 7.2.

Gmsh library was used to create the 3D finite element grid. The steps required to create an OpenFoam compatible mesh with Gmsh library and its results are explained below with Figures X to Y supporting the explanation.

1. First step is to create a ".dat" file with 3 column format X Y and Z coordinates for the airfoil. The airfoil database: [19] provides the X and Y coordinates for the airfoil profile. In order to make the file compatible with Gmsh requirements, a third column separated with simple spaces and same number of rows as the previous ones but full of zeros is enough. That means that our airfoil is a 2D surface. Airfoil ".dat" file is shown in Section 7.1.

2. The second step is to make this "dat" file compatible with Gmsh displaying options. Unlike OpenFoam, Gmsh has a graphic interface that looks like Figure 16

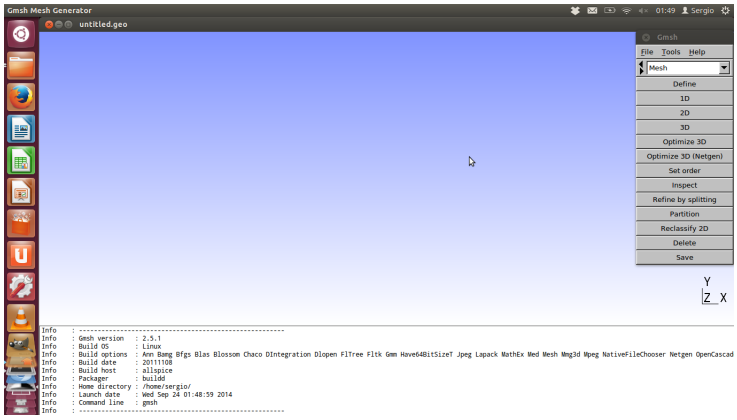


Figure 16: Gmsh graphic interface

. For this purposes, a converter Python file is required. The converter file was obtained from [20]. After running the converter file with the command shown below, the resultant output

is swhon in Figure 17.

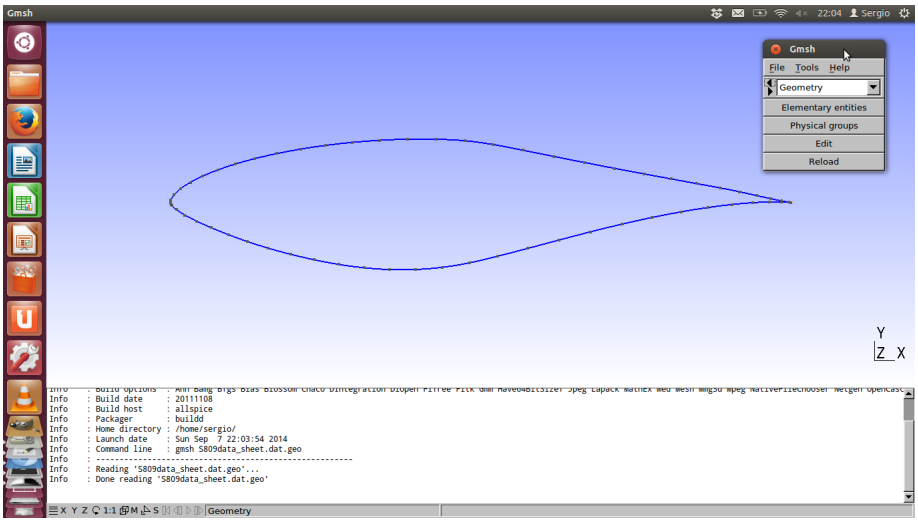


Figure 17: Airfoil profile mesh

```
python dat2gmsh.py S809data_sheet.dat
```

3. The result of this operation is the desired airfoil profile. Next step is to add the environment in which our airfoil will stand. For the purposes of this project, a rectangular box with size 30c (where "c" is the cord of our airfoil) was chosen. The distribution of this length is 10c upstream and 20c downstream. For upper and lower separation, 10c distances where selected. The sizing of the environment surrounding the airfoil is set by adding the four corner points and linking them with lines. The commands added to the ".geo" file created before are:

```
edge_lc = 0.2;
Point(1100) = { 20, 4, 0, edge_lc};
Point(1101) = { 20, -4, 0, edge_lc};
Point(1102) = { -10, -4, 0, edge_lc};
Point(1103) = { -10, 4, 0, edge_lc};
Line(1) = {1100,1101};
Line(2) = {1101,1102};
Line(3) = {1102,1103};
Line(4) = {1103,1100};
// Create a boundary and define airfoil Spline
Line Loop (1) = {1,2,3,4};
Line Loop (2) = {1000};
// Create surface and Volume
Plane Surface(1) = {1,2};
Physical Volume("internal") = {1};
```

The result of that operation is shown in Figure 18. This commands are required for every single mesh that is created, i.e. for every single case that is going to be run.

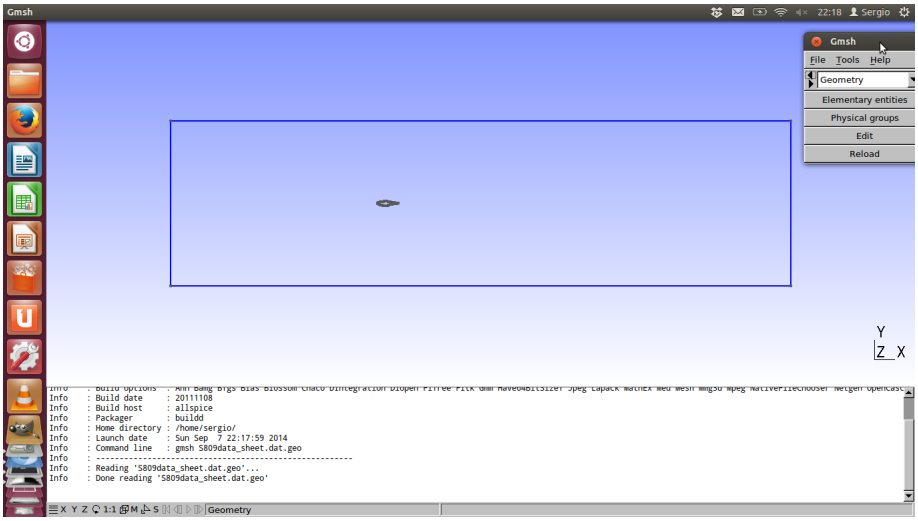


Figure 18: Mesh Contour definition

4. After both, the airfoil and the boundaries of the environment, are defined, the last step with Gmsh is to create the cells that will define the mesh. To perform this operation, the first thing users may be aware of is to extrude the section in order to make the output file compatible with OpenFoam.[3] For extruding the surface and making file fully compatible with OpenFoam, we require the extruding operation and surfaces+volume definition. The following commands should be added to the ".geo" file that we are working with:

```
Extrude {0, 0, 1}
{ Surface{1};
Layers{1};
Recombine;
}
Physical Surface("back") = {1027};
Physical Surface("front") = {1};
Physical Surface("top") = {1022};
Physical Surface("exit") = {1010};
Physical Surface("bottom") = {1014};
Physical Surface("inlet") = {1018};
Physical Surface("airfoil") = {1026};
Physical Volume("internal") = {1};
```

Note: it is very important to add all these commands to our "S809data_sheet.geo". It does not change the geometry of the mesh but makes the resulting file compatible with OpenFoam. It is important to remark that the surfaces created with the commands shown in the previous box are the different surfaces that will comprise the final mesh of our profile. In order to have a clear understanding of which are the different surfaces, refer to Figure 19.

[3] Even if users are dealing with 2D profiles, OpenFoam requires a 3D meshing of at least 1 cell depth to run the cases.

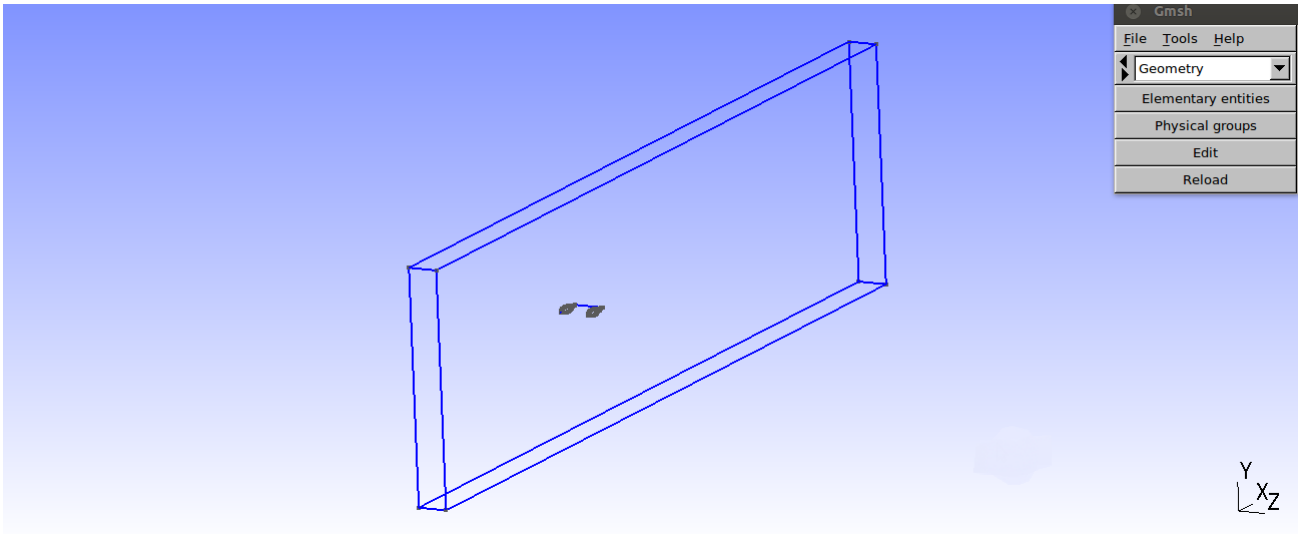


Figure 19: Mesh Surfaces

5. After this file edition, there is just a simple operation remaining: Just by clicking the menu that was shown in Figure 16, selecting "Mesh" tab and choosing 3D. By clicking on that button, Gmsh will automatically show the resulting mesh as in Figure 20.

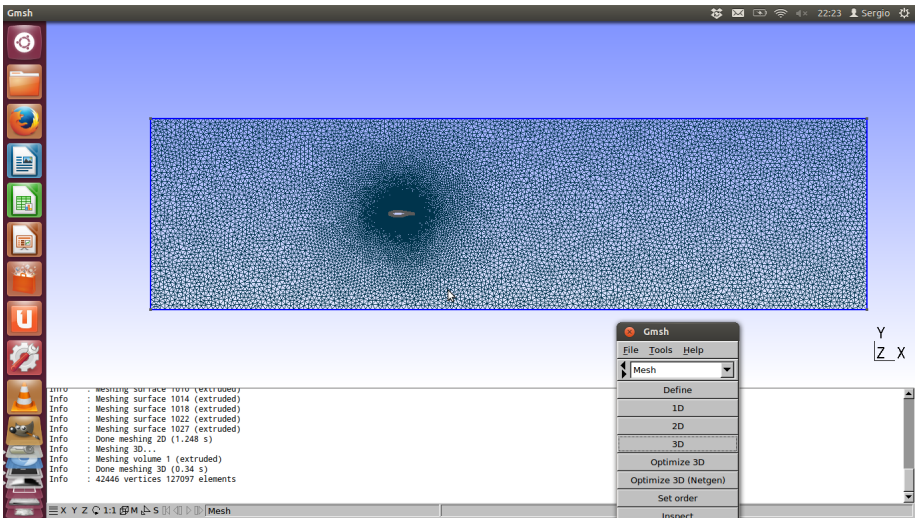


Figure 20: Final Mesh

In terms of accuracy and reliability of this mesh, in the following pictures, further views of the mesh show the fact that as we approach the critical points of the airfoil: mesh has a higher density of cells in Leading and Trailing edges and the surrounding areas because these are the points where the higher pressure gradients are found as will be shown in the c_p graphs provided by XFLR5.

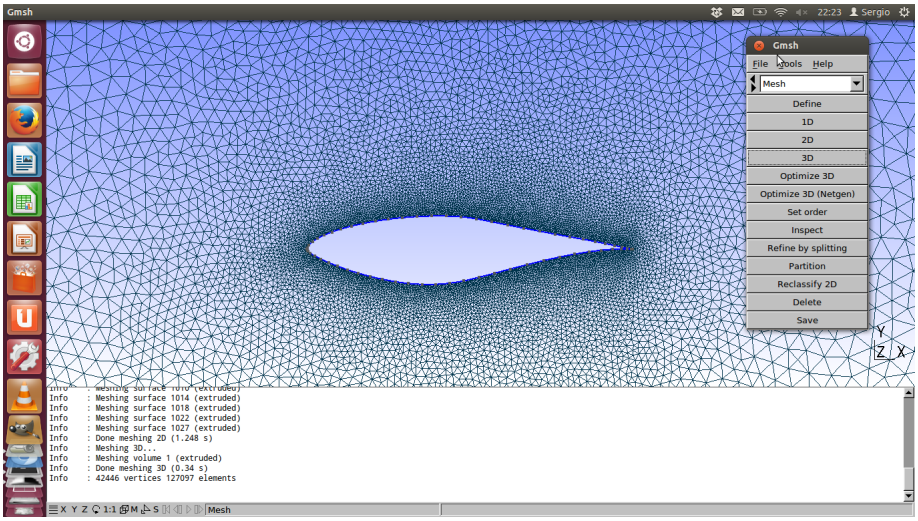


Figure 21: Mesh Airfoil Detail

Figure 21 shows the distribution of cells on further regions of the airfoil and on regions very close to the airfoil.

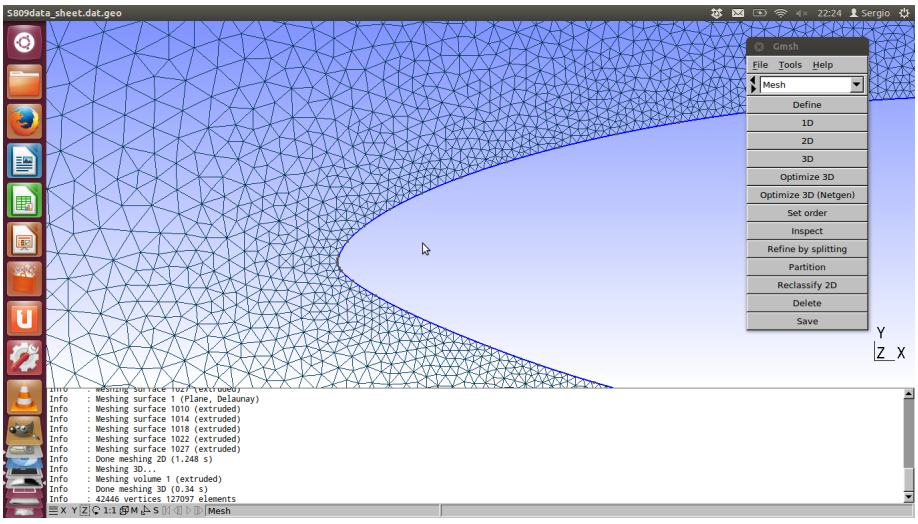


Figure 22: Mesh Leading Edge Detail

Figure 22 shows the size of the cells closer to the leading edge of our airfoil. Small size of the cell will provide improved accuracy on the results, because the pressure variations among cells that are in contact will be small and accurate graphics can be plotted when more data points are provided.

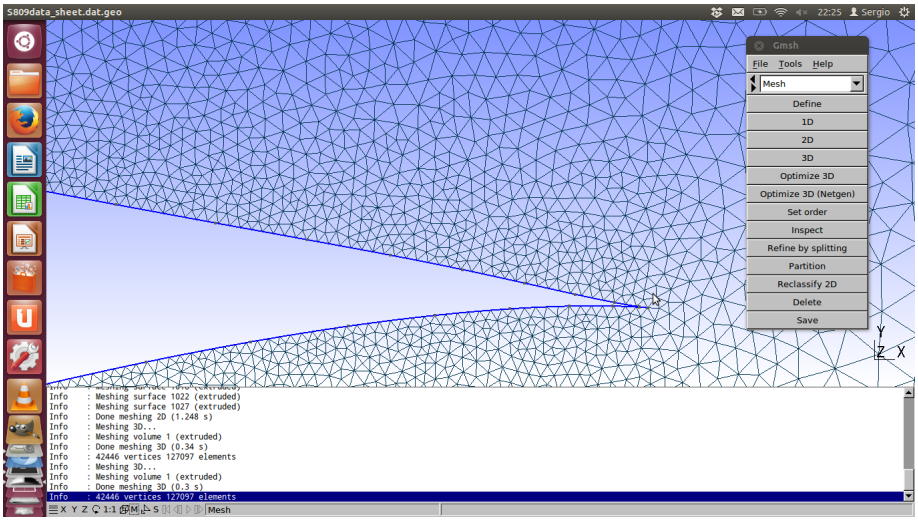


Figure 23: Mesh Trailing Edge Detail

In Figure 23 a similar scenario as in Figure 22 is shown. Leading edge region is critical for boundary layer separation and, therefore, more data cells are required in this region to provide an accurate estimation of the separation point.

3.1.4 Mesh validation

In order to validate the results of the mesh that is being used for this project, it is required to estimate the size of the cells. Increasing the number of cells (reducing cell size) will provide more accuracy to the results, in order to optimize the resources, an optimized geometry obtained by Gmsh software is used. Gmsh tool provides user a tool for mesh refinement. By using this tool and running a simulation with the improved mesh, the validity of the results can be confirmed. If the results of the OpenFoam simulation of the original and the refined mesh are reasonably similar, the original mesh is considered to be valid in order to reduce the computational costs. For this project, the original mesh for the Scenario 2 airfoil is shown in Figure 24.It has 123123 elements and after redefinition of the mesh by using the "Refine by Splitting" tool twice, a final number of 652859 elements was achieved.

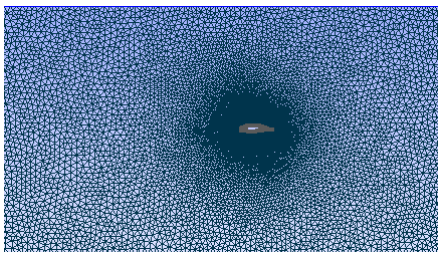


Figure 24: Original Mesh Scenario 2

The final redefined mesh is shown in Figure 25 and it can be easily appreciated that the cell density has substantially increased. The results for the mesh validation do not present relevant variations to those from the original ones. Using 6 decimals of accuracy, the changes were only

appreciated on the 4^{th} digit and, therefore, they were always on the order of ~0.1% maximum. That implies that the initial mesh is reliable and that it is not worth to spend computational resources on improving the mesh to that level of accuracy.

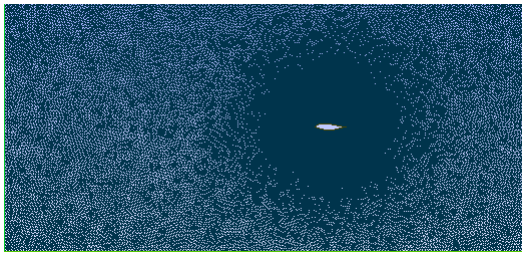


Figure 25: Refined Mesh Scenario 2

3.2 XFLR5

3.2.1 Description

According to the definition provided by the official website of XFLR5 [8]: "XFLR5 is an analysis tool for airfoils, wings and planes operating at low Reynolds Numbers. It includes: XFoil's Direct and Inverse analysis capabilities Wing design and analysis capabilities based on the Lifiting Line Theory, on the Vortex Lattice Method, and on a 3D Panel Method."

According to this definition, different concepts need to be explained:

- Direct Analysis: analysis in which the operations begin with loading the airfoil geometries and the different operations are performed to this airfoil in order to obtain the desired polars, pressure distribution, or evolution of different parameters with respect to AoA, Re or Mach number. The simplified steps (a more precise case set-up example is given in Section 4.1.1) in order to perform this type of analysis according to the official XFLR5 user guide are:
  - Foil object/s are loaded with the airfoil data starting at the trailing edge, rounding counter-clockwise the airfoil and reaching the leading edge before going back to the trailing edge. The data format for the coordinates of the airfoil should be of the form: X(1) Y(1) X(2) Y(2)... X(N) Y(N), where X data is placed in one column and Y data is placed on a different column without comas, only a simple spacing is required. Files that do not fulfill this format will generate errors during the loading of the profile. Usually 50 points for each, upper and lower, surface of the airfoil are enough to obtain an accurate analysis. More data points implies an improved accuracy but there is a limit of 150 points per surface; once that limit is reached, no additional accuracy can be obtained.
  - Analysis: after selecting the desired conditions for Re, Mach and AoA; the actual analysis can be run. There is also the chance to select the laminar to turbulent transition.
  - Preview of the results: when the analysis is performed, XFLR5 provides different options for the graphic results to be displayed on its control panel. Users are allowed to choose among the different results and select a particular Re, Mach and/or AoA condition to be displayed.
- Inverse Analysis: by importing the c_p distribution, XFLR5 is able to generate the airfoil that corresponds to that pressure distribution once the α and Re conditions are provided. Further explanation is provided in the Section 4.1.1.

- Foil design:
  - XFLR5 offers the option to create a desired airfoil by editing a default geometry. User is allowed to change the shape by just clicking on the surface and moving the default points to the new desired position. However, this is not the best way to design profiles because the resulting profiles are not usual. It is only recommended for trials and when the user does not have a desired purpose.
  - Numeric modification of camber and thickness is a second choice of creating the desired profile. This option offers more accuracy on the shape of the final design. This is recommended when different profiles with similar shapes want to be loaded and analyzed simultaneously for comparison among them.
  - Loading existing profile data file: this is the recommended procedure because it provides full control and accuracy. Users can obtain the airfoil data files in many different sources and just by editing these files to fulfill the requirements explained before, the analysis will be available in a few seconds.

3.2.2 Operations with XFLR5

For the purposes of this project, the following operations will be performed in XFLR5:

1. Airfoil data import from data file obtained from [19]. This data file is provided in section 7.1.
2. Direct Analysis selecting the given Re and Ma at which the analysis will be performed.
3. α range selection according to the purposes of this project.
4. Polar and pressure distribution graphs and data export.

This operations will be performed in the first stage of the optimization process explained in Section 2.2. After that, OpenFoam analysis will be performed. Once the analysis with OpenFoam is completed, the results for the original airfoil analyzed with the two different software will be compared. After that, using Inverse Analysis capabilities of XFLR5, a new airfoil with improved characteristics will be created. The new airfoil data will be compared with the original airfoil data obtained from XFLR5 and, again, new airfoil will be analyzed with OpenFoam for further comparison with the results of the original airfoil.

After that, if any improvement or additional optimization of the airfoil is required, new pressure distribution data will be created and the operations described on the previous paragraph will be performed.

The airfoil data obtained out of the Inverse Analysis, will be compared with original airfoil data in order to provide a graphic explanation on how the profile shape improvement reduces drag and delays separation point conserving the area of the profile in order to maintain the material required to manufacture the profile and avoid an increase on manufacturing costs.

3.3 OpenFoam

3.3.1 Description

According to the definition provided on the official website of OpenFoam: "OpenFoam (Open source Field Operation And Manipulation) is a C++ toolbox of numerical solvers and the related pre- and post-processing libraries required for the complete analysis of continuum mechanics, in

particular CFD problems. Its development started in the 1980s at the Imperial College in London. OpenFoam runs only on Linux operating system computers.

The most distinguishing feature of this software is its intuitive syntax. Thanks to the flexibility this syntax offers, users can develop their own customized solvers on a relatively easy manner. All the conditions required for the solution of the problem can be modified by just editing the input files that the solver requires. OpenFoam allows user to solve a wide range of problems: from linear systems and ordinary differential equations to complex thermodynamic problems or turbulence models. "

3.3.2 Why OpenFoam?

The selection of OpenFoam as software for this study was made based on three main factors:

The first consideration is that CFD software is complex, besides that, and as a second reason. commercial CFD available licenses are expensive. Finally, this type of open-software has more powerful capabilities due to the fact that a whole community is working and improving OpenFoam's core and applications with evident advantages.

This project aims to test OpenFoam capabilities by comparing it with a widely used software: XFLR5 (See section 3.2 for more details on XFLR5). Some of the advantages of OpenFoam that led to the selection as reference software for this project are:

- Wide range of applications updated by user community
- No license cost
- Support by community developers
- Friendly syntax

On the other hand, OpenFoam has its own drawbacks: since it is comprised by numerous solvers and additional applications, it is hard to develop an standard graphic interface and additional post-processors are required to display the results. This is the main disadvantage but once the learning curve peak is reached, users acquire the knowledge to perform the different required operations with basic Linux commands. In order to make it easy for any reader who is willing to reproduce the results of this paper, in Section 8 a tutorial on how to use OpenFoam and run a case is provided.

3.3.3 Operations with OpenFoam

In this section, all the procedures that were performed in order to achieve the stated goals in Section 2.1. Analysis with OpenFoam require three main steps:

1. Pre-processing: in this stage, all the different documents required for the case set up are prepared and filled using different pre-processing software (E.G: gmsh package for mesh generation). Description of the different files required for OpenFoam analysis is provided on Section 8.4.
2. Running the analysis: different solvers can be used with OpenFoam according to the different outputs that the user expects from the analysis. Some examples are solvers simpleFoam and icoFoam. For the purposes of this paper, solver icoFoam which solves for incompressible, laminar flow of Newtonian fluids.
3. Post-processing: by using ParaView block that was already installed with OpenFoam as it is shown in Section 8.2, the graphic desired results can be achieved

This three main blocks contain all the operations that will be explained with further detail on Section 4.2.1 OpenFoam Case Set-up

4 Software

4.1 XFLR5

4.1.1 Case setup

In this section, the steps required to set-up a case in XFLR5 will be shown. This section provides the required steps for the reader in order to reproduce the whole procedure in which this paper is based.

1. Creating a new project/Opening existing project: XFLR5 offers the option of using existing data files for airfoils and import them to start a project with the desired airfoil. The condition is to use a .dat file where the data points of the airfoil start in the trailing edge and move clockwise up to the leading edge and then back to the trailing edge through the upper surface of the airfoil. In this project, S809 airfoil data was loaded as it is shown in Figure 27.

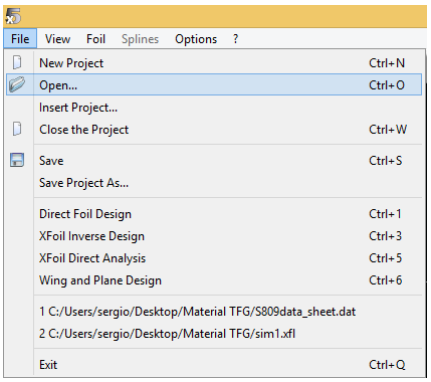


Figure 26: Opening project XFLR5

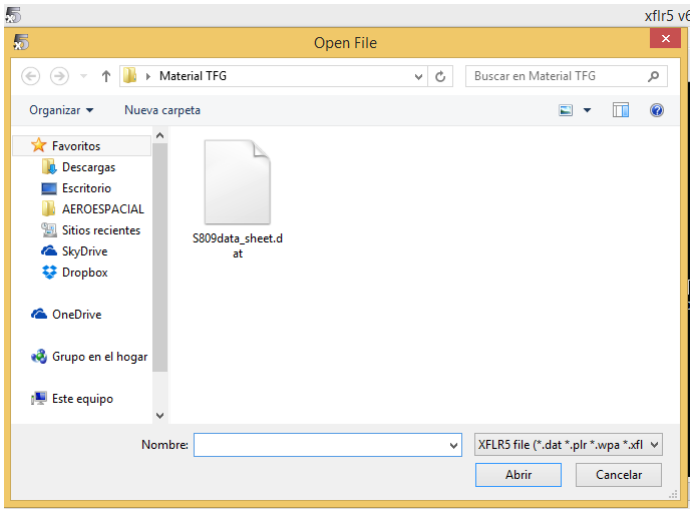


Figure 27: Loading S809 data file

2. After the airfoil data is loaded, the user will see the actual profile. For convenience, user shall Rename the airfoil with the desired name.

3. Once the airfoil profile is loaded, the type of Analysis needs to be selected. The different types of analysis that can be performed with XFLR5 were shown in Section 3.2.2. For the purposes of this project, the first analysis that will be performed is a "Direct Analysis".

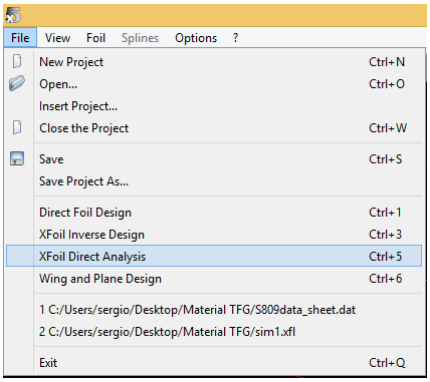


Figure 28: Direct Analysis for airfoil S809

4. The next screen user will find is the control panel for the direct analysis. In this screen, all the different plots that can be obtained from XFLR5 Direct Analysis are shown. They are empty graphs because no analysis as been performed yet.

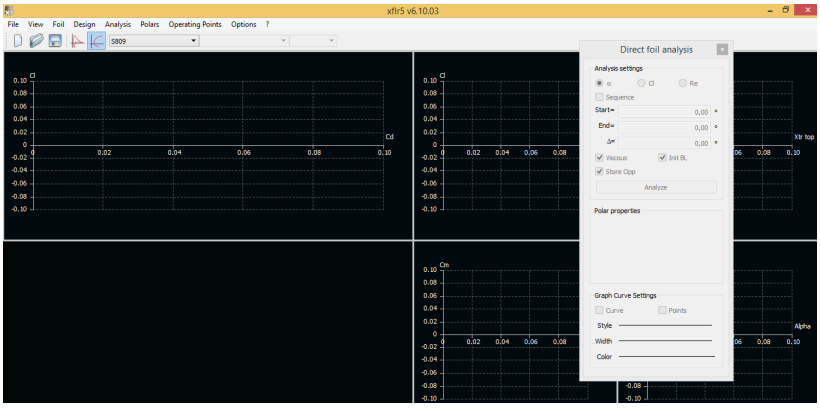


Figure 29: XFLR5 direct analysis control panel

Next step is to open the Analysis Tab button and select Batch analysis.

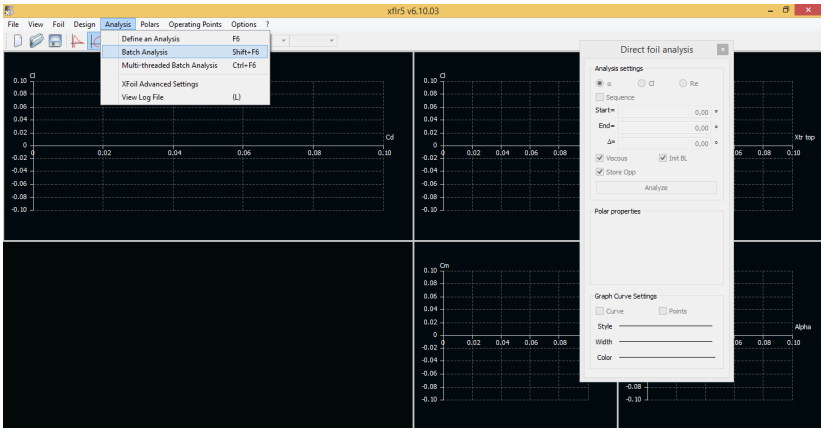


Figure 30: Batch Analysis selection

5. User would be allowed to define his own analysis but for the purposes of this project, one of the XFLR5 predefined analysis will suit the required specifications. As it is shown in Figure31, the choice for this project is a Type 1 analysis. For Type 1 analysis, the configuration selected for the purposes of this simulation was: $Re\epsilon[950.000, 1.000.000]$, $Ma = 0.15$, $Ncrit = 9.0$ and $\alpha\epsilon[-5^{o}, 20^{o}]$.

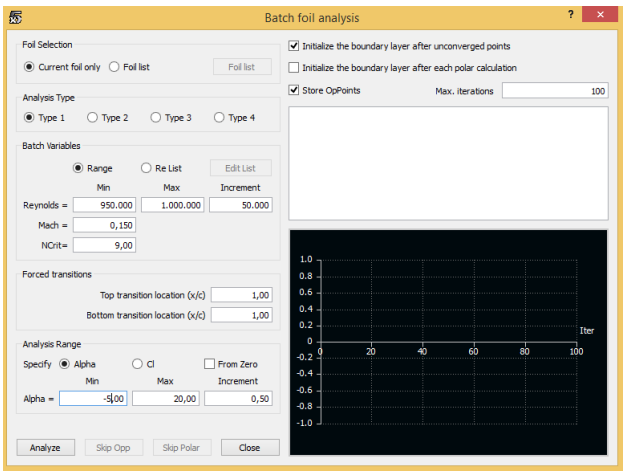


Figure 31: Batch Analyis type selection

6. After clicking "Analyze" button, the analysis will begin with the parameter selection explained on the previous paragraph. Once the analysis is completed, XFLR5 will show again the control panel but this time, the graphs will present the results of the analysis. These results are shown in Figure 32. User can easily edit the layout of the graph by double-clicking on it and selecting the parameters that will be shown on the x and y-axis respectively.

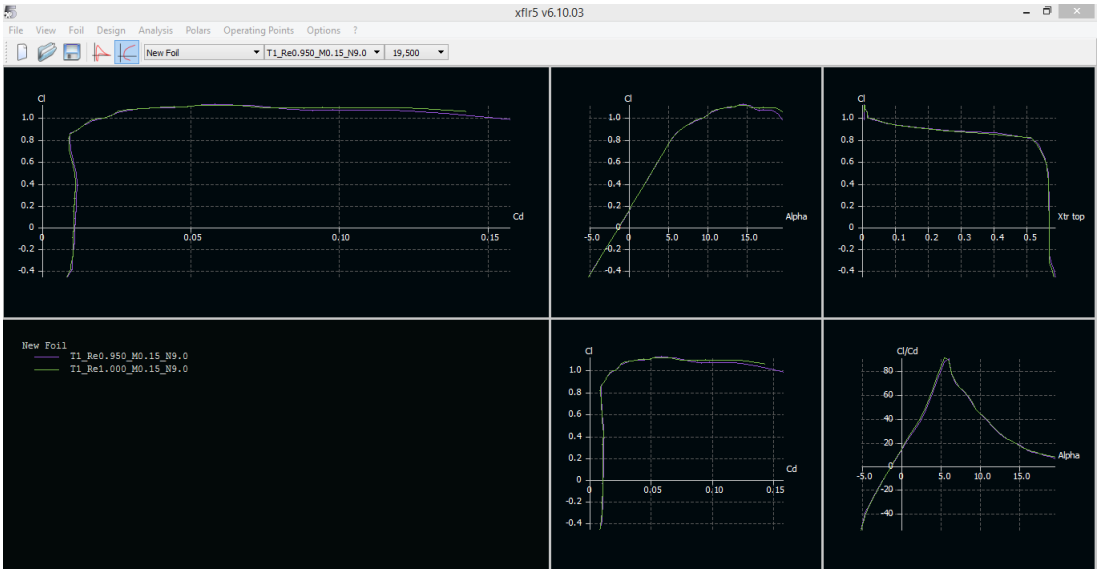


Figure 32: Graphic results for XFLR5 analysis

7. Finally, and this is probably the most useful result that XFLR5 provides for the purposes of this study, the c_p distribution of the analyzed airfoil is shown by just clicking on the button right on the left of the blue-shaded one on Figure 32on the upper tab of the program. This button has a schematic view of an actual c_p distribution, providing an intuitive user experience. The results of the analysis with the desired c_p distribution are shown in Figure 33. User has the option to select among the different c_p distribution according to the different angles of attack that were selected on step 5 of this list.

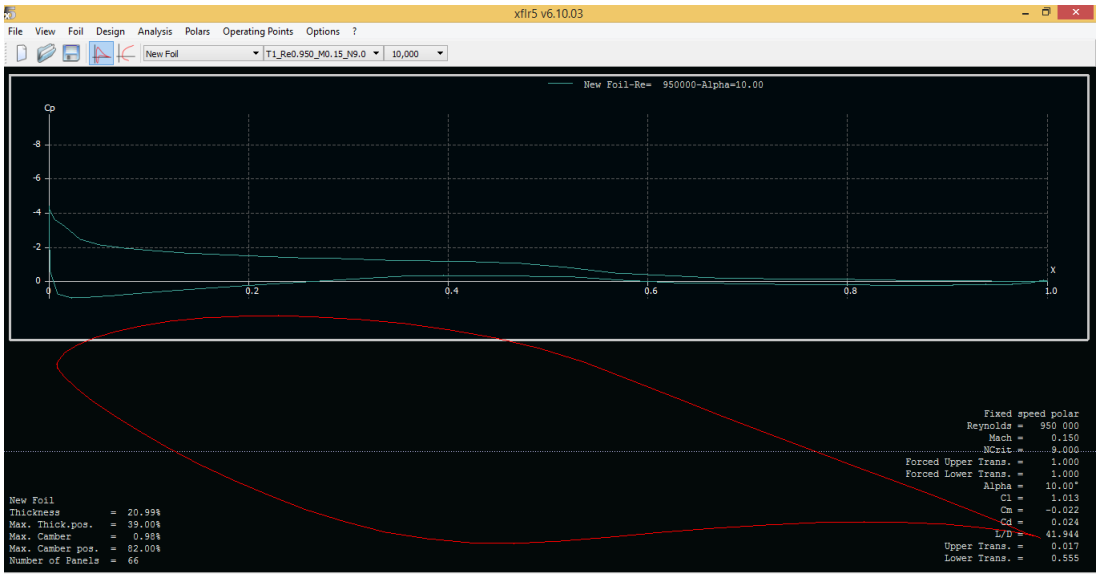


Figure 33: Pressure distribution XFLR5

Inverse design: in order to perform the desired Inverse Analysis to obtain the new airfoil with improved aerodynamic characteristics, the following steps shall be followed:

1. Switch to the Full Inverse Analysis in the main control tab:

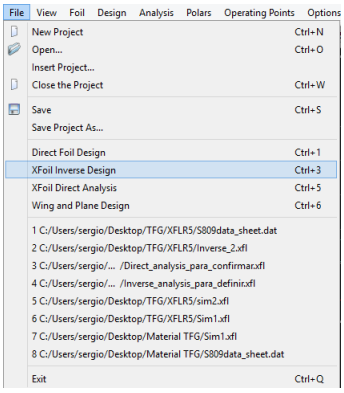


Figure 34: Inverse Analysis selection

2. Select the original S809 airfoil in order to have the reference airfoil on screen.

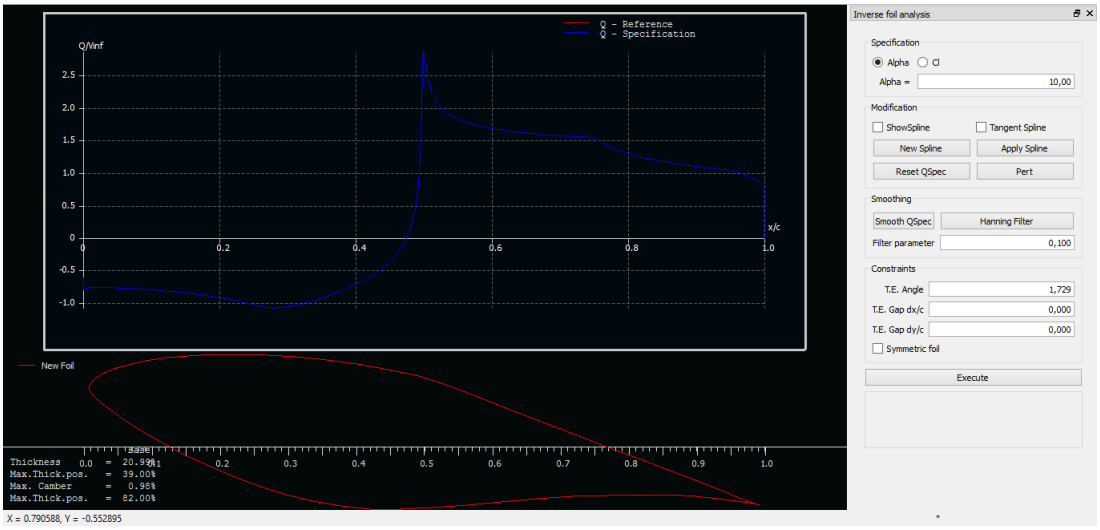


Figure 35: Inverse Analysis Original Airofoil

3. Select the "New Spline" button in the toolbar as shown in Figure 36.

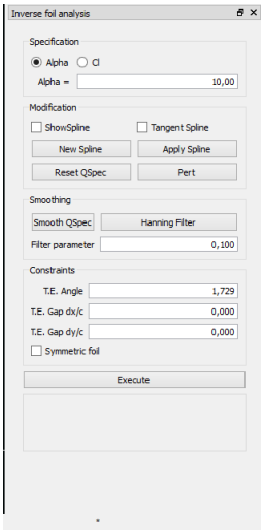


Figure 36: New Spline

4. Select two points both on the upper or lower surface, but not one on each. This is important because the pressure distribution will be optimized in the region that is contained within these two points. As it is shown in Figure 37, the area of main interest is where there is a pressure peak, that implies that the flow has separated in that area and that is the region to be improved.

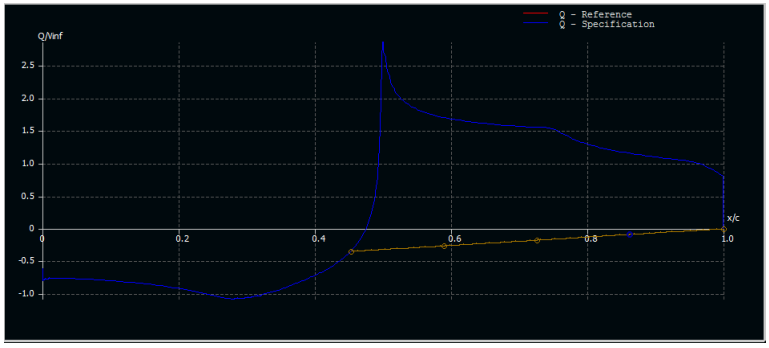


Figure 37: Optimization Region Selection

5. Drag the control points to define a new distribution for the desired region. Smooth distributions with no peaks imply a reduction on drag and a delay on separation. It is important, for purposes of aerodynamic efficiency improvements, to keep the area below the curve as similar as possible or try to increase it in order to obtain more lift without increasing drag by making the shape of the speed distribution smooth as it is shown in Figure 38.

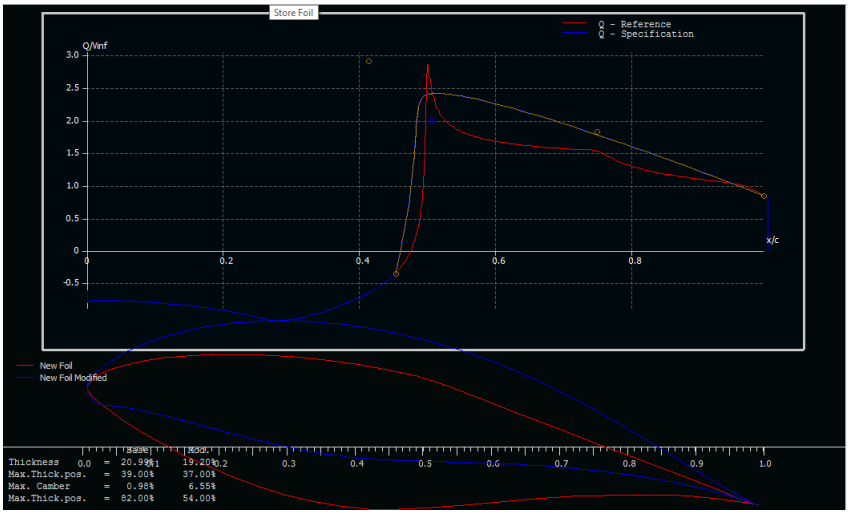


Figure 38: New airfoil

6. Click on the "Apply" and “Execute” in order to save the changes and calculate the new geometry.
7. Repeat steps 3-6 until a reasonable or desired shape is achieved.
8. In order to save the new airfoil geometry, click on the arrow in the top toolbar, or select "Store foil in the database".

Figure 39: Storing new profile

9. To perform the Direct Analysis on this new airfoil, switch to the Direct Analysis as it was shown in previous steps. It is important to “Refine” the profile in order to generate a suitable data file that XFLR5 can run with the panel method.

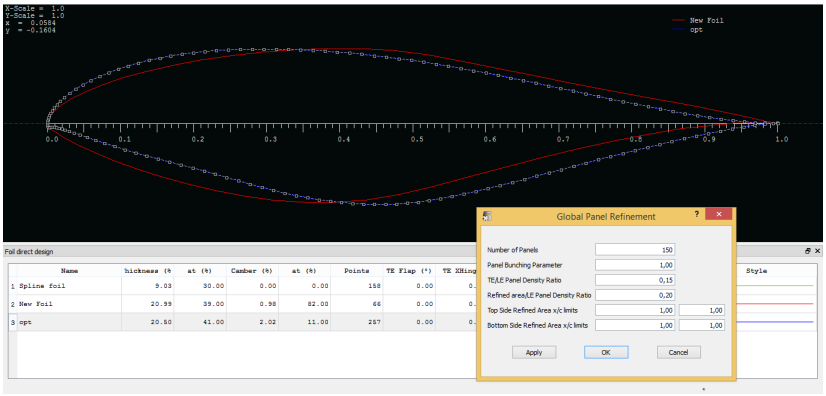


Figure 40: Profile Refinement

10. Proceed with the direct analysis with the same conditions that were established on the analysis for the original airfoil.

4.1.2 Objectives

Case set-up explained on Section 4.1.1 has the purpose of obtaining the desired results and appropriate outputs in order to achieve the goals of this report. Concerning XFLR5, the main goals are listed below:

1. The first goal with these operations is to obtain the polars of S809 airfoil for a given α and Re.
2. The second goal is to obtain c_l distribution. This output will be used for comparison with the results obtained with OpenFoam.
3. Analysis of the stall behavior of the airfoil by studying c_l vs. α graph.
4. Estimation of separation point out of c_p distribution.

4.2 OpenFoam

4.2.1 Case setup

In the following images, the step by step case setup is shown with a brief explanation of each of the steps and the parameter selection according to its rationale.

1. The first step is to convert and check the mesh in order to "activate" it so OpenFoam can run the case based on that mesh. In order to do that, just activate the case directory on the terminal and type the following commands:

```
$ gmshToFoam NAMEOFTHEMESH.msh
$ checkMesh
```

2. It is required to go through each of the folders and make the desired changes according to the conditions of the problem to be solved. For the purposes of the project, the first required change is within the file Constant/polyMesh/boundary on the case directory. In this file and according to the different surfaces that were created for the mesh, the type of each surface must be properly defined. For this project, "front" and "back" surfaces must be of type "empty" on that file, and "airfoil" must be a type "wall".
3. The next folder where some changes need to be performed is on the initial condition folder named "O" in the case directory. In this folder, the initial conditions for velocity, speed and viscosity effects are set up. In the following pages due to the size of the scripts, the command boxes that shows how each of the files should look like are shown:

(a) P file:

```
inlet
{
type freestreamPressure;
}
exit
{
type freestreamPressure;
}
top
{
type freestreamPressure;
}
bottom
{
type freestreamPressure;
}
airfoil
{
type zeroGradient;
}
front
{
type empty;
}
back
{
type empty;
}
```

(b) U file: (where U_x,$U_y$and U_Z are the components of the free stream velocity for the case to be run)

```
inlet
{
type freestream;
freestreamValue uniform (Ux Uy Uz);
}
exit
{
type freestream;
freestreamValue uniform (Ux Uy Uz);
}
bottom
{
type freestream;
freestreamValue uniform (Ux Uy Uz);
}
top
{
type freestream;
freestreamValue uniform (Ux Uy Uz);
}
airfoill
{
type fixedValue;
value uniform (0 0 0);
}
front
{
type empty;
}
back
{
type empty;
}
```

(c) nut file: (where 0.14 is the parameter set up for this case and represents the level of distortion upstream of the profile)

```
inlet
{
type freestream;
freestreamValue uniform 0.14;
}
exit
{
type freestream;
freestreamValue uniform 0.14;
}
top
{
type freestream;
freestreamValue uniform 0.14;
}
bottom
{
type freestream;
freestreamValue uniform 0.14;
}
airfoil
{
type nutUSpaldingWallFunction;
value uniform 0;
}
front
{
type empty;
}
back
{
type empty;
}
```

(d) and the nuTilda file:

```
inlet
{
type freestream;
freestreamValue uniform 0.14;
}
exit
{
type freestream;
freestreamValue uniform 0.14;
}
top
{
type freestream;
freestreamValue uniform 0.14;
}
bottom
{
type freestream;
freestreamValue uniform 0.14;
}
airfoil
{
type fixedValue;
value uniform 0;
}
front
{
type empty;
}
back
{
type empty;
}
```

4. After this process, the case is ready to be run by just typing the name of the solver that is going to be used. In this case the selected solver was "simpleFoam"

5. Once the case is completed and all the time steps have been completed, by using "paraFoam" command, the results can be analyzed.

4.2.2 Objectives

For the purposes of this project, the desired objectives after OpenFoam operations are:

1. Provide readers with a tutorial on how to use OpenFoam for airfoil analysis and generate value to the aerodynamic community by opening a new software that may allow users to perform their analysis without purchasing the expensive licenses associated with other CFD software.

2. Validate software reliability by comparing results with XFLR5 polars

3. Obtain pressure distribution of S809 airfoil.

4. Obtain a graphic interpretation of the separation point of S809 airfoil.

5. After improved airfoil shape is created with XFLR5, obtain pressure distribution and separation point of new airfoil.

These are the main objectives, further details on each of them are provided in the results Section 5.2.

5 Results

This results will be referred to the objectives that were set up on Section 4.1.2 and Section 4.2.2. All the results will be focused on the list of objectives provided by these sections. It is important to remark that the tolerance error for OpenFoam calculations was 10^{-8}.

5.1 XFLR5 outputs

From XFLR5 Direct Analysis, the relevant outputs are: S809 airfoil polar (c_l vs. c_d) and the evolution of lift coefficient with respect to the angle of attack to obtain the critical angle of attack that implies stall and generally has assciated the presence of boundary layer separation. Some additional and supporting plots such as c_{dp}evolution and pressure distribution for the selected angle of attack (10^{o} throughout the whole project) will be provided because they will be relevant for the conclusions that can be obtained out of this project.

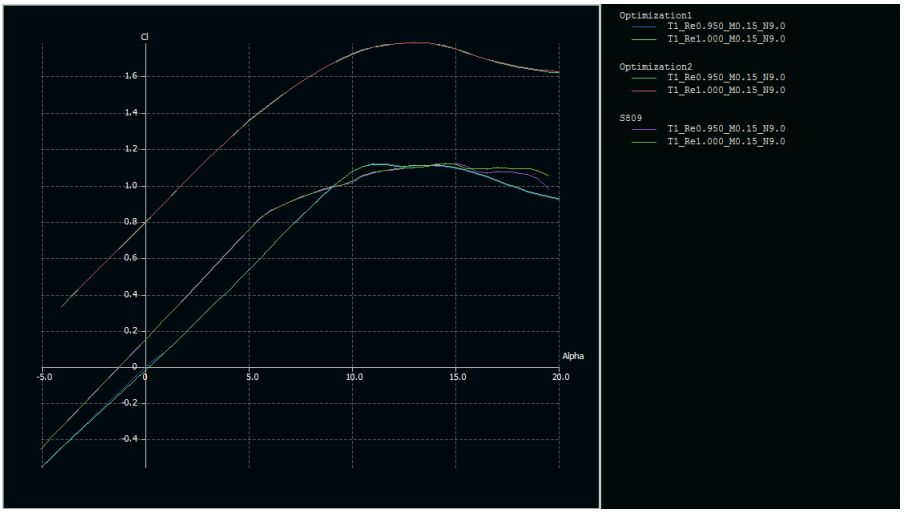


Figure 41: cl vs α comparison

This is probably the most ilustrative figure of the optimization method that was performed. It is appreciated for the original S809 airfoil that the stall point is around 12.5^{o}. The evolution of the lift coefficient presents its traditional linearity for low angles of attack until reaching the maximum lift coefficient. Whereas the lift coefficient has decreased until 10.5^{o} for the first optimization process, an improved performance on boundary layer separation has been achieved. That is due to the fact the the airfoil presents a different curvature on the lower surface. On the other hand, the second optimization has improved the evolution of lift coefficient not only by increasing camber and thickness (shifting the line upwards), but also by improving the boundary later separation with a maximum lift coefficient obtained at 13.5^{o}. That delay on stall implies that the separation point has been delayed a sufficient amount for the airfoil to be able to pitch up $+1^{o}$ without generating turbulence and stalling.

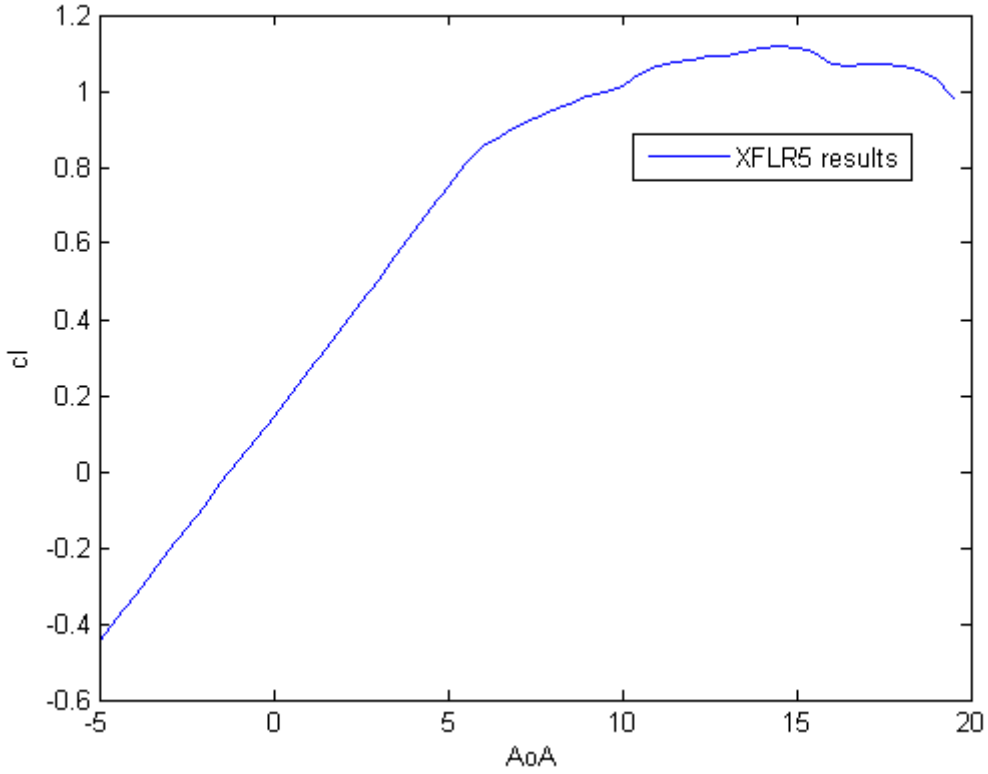


Figure 42: S809 c_l vs. α

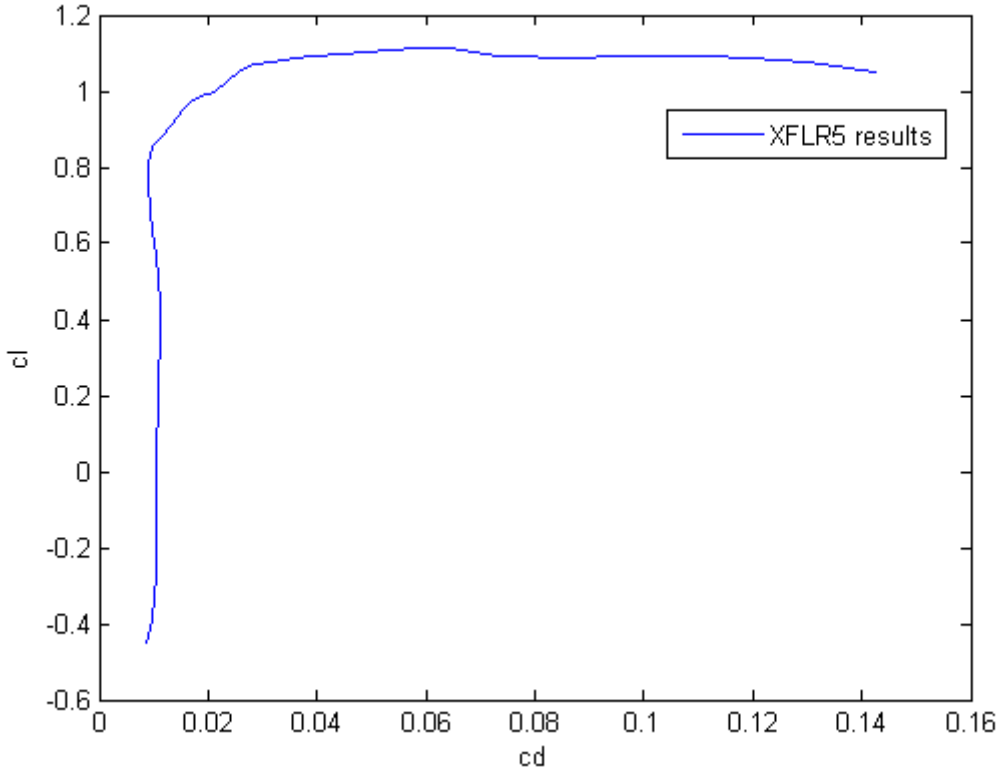


Figure 43: S809 polar

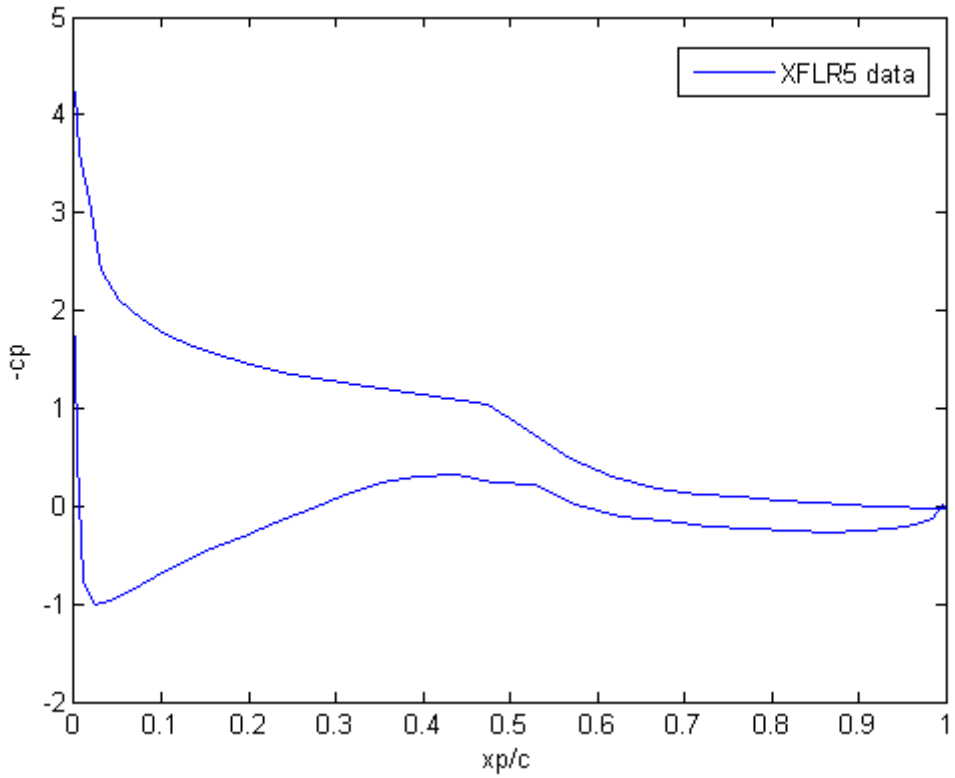


Figure 44: S809 Pressure Distribution at 10^{o}

All the Figures shown before are just the baseline of the graphs of interest for this project. It is interesting to mention that the stall region where boundary layer separates for the original airfoil is in the neighborhood of 15^{o} and that the separation point is on position $x = 0.47\bar{c}$. Regarding the Inverse Analysis, the new profiles and the graphic results mentioned above will be shown on dedicated Sections 5.3.1 and 5.3.2 for each Scenario.

5.2 OpenFoam outputs

The OpenFoam outputs will be presented for each of the three airfoils analyzed: the original S809 airfoil, the optimized airfoil for improved c_{dp} (Scenario 1) and the optimized airfoil for aerodynamic efficiency (Scenario 2).

- Pressure distribution:

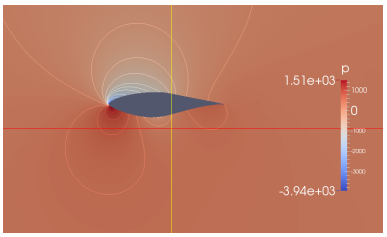


Figure 45: OpenFoam S809 pressure distribution

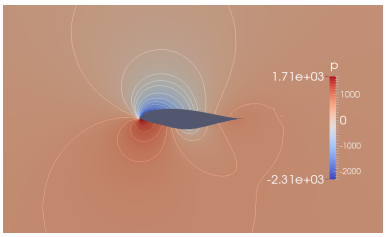


Figure 46: OpenFoam Scenario1 pressure distribution

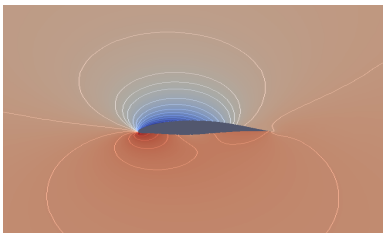


Figure 47: OpenFoam Scenario2 pressure distribution

It can be appreciated that the pressure distribution near the trailing edge varies from one scenario to another. That is the result of the optimization process and the delay on boundary layer separation. There are also two interesting remarks that will be appreciated on the pressure coefficient comparison with XFLR5 results: on Scenario 1, there seems to be a recirculation bubble on the lower surface of the profile. That bubble is caused because on that region, the curvature of the profile changes and the pressure gradient is affected by the shape of the airfoil. The second remark is that on Scenario 2, there is a small area on top of the upper surface where the pressure is decreased and that might be caused by partial separation or recirculation on that area with potential risk of bubble formation.

- Velocity distribution:

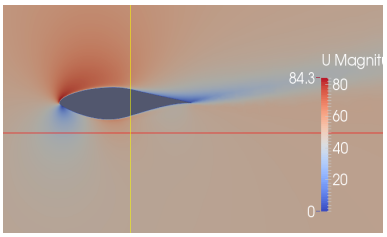


Figure 48: OpenFoam S809 velocity distribution

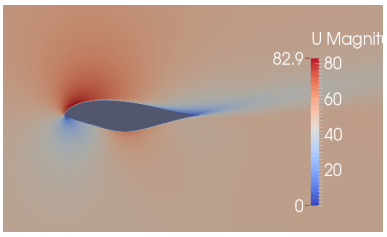


Figure 49: OpenFoam Scenario1 velocity distribution

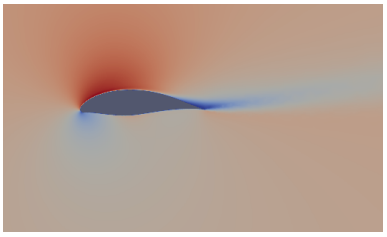


Figure 50: OpenFoam Scenario2 velocity distribution

As it was expected, the figures show that there is acceleration of the flow on the upper surface of the profile and deceleration on the lower and trailing edge of the profile. OpenFoam provides a very intuitive display that reinforces the expected results.

- Flow Streamlines:

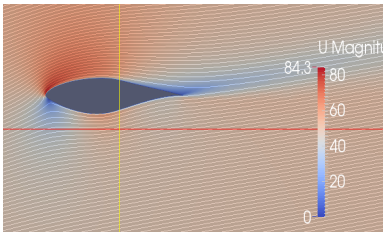


Figure 51: OpenFoam S809 streamlines

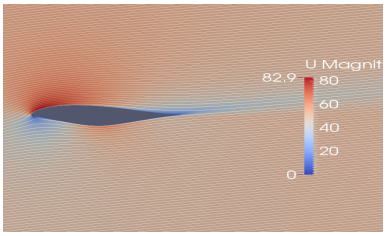


Figure 52: OpenFoam Scenario1 streamlines

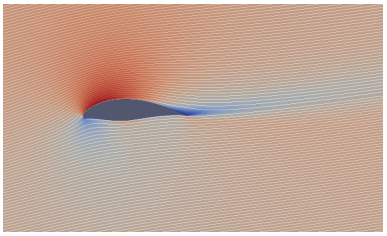


Figure 53: OpenFoam Scenario2 pressure distribution

This figures are providing the information that was expected: the flow is laminar and not very disturbed on the environment besides the region where streamlines diverge on the leading edge and the separation region near the trailing edge; the angle of attack is not big enough to produce separation bubble even though the turbulent boundary layer detaches from the airfoil at the positions that are estimated in Section 5.4.

- Lift and Drag coefficients for different angles of attack: making the appropriate choices for angles of attack, these table shows the values of the lift and drag coefficient for each of the airfoils obtained from OpenFoam. These results will be plotted together with the XFLR results in Section 5.4

	S809		Scenario1		Scenario2	
α	c_d	c_l	c_d	c_l	c_d	c_l
0^o	-0.0140806	0.143564	-0.0166477	0.0649281	-0.022884	0.669935
5^o	0.0299983	0.634306	0.0191359	0.44146	0.0644945	1.1407
10^o	0.1202	0.99262	0.110506	0.869927	0.196382	1.45438
15^o	0.178245	1.10931	0.186089	1.05625	0.0324243	1.33962

Table 1: Coefficients data OpenFoam

5.3 Optimization method

Data of each airfoil after optimization process obtained from XFLR5 results:

Airfoil	Max. Thick (%)	At (%)	Max Camber (%)	At(%)	Points
S809	20.99	39.00	0.98	82.00	66
Opt1	20.50	41.00	2.02	55.00	65
Opt2	19.69	38.00	5.87	11.00	65

Table 2: Airfoil optimization process data

5.3.1 Scenario 1

Using Inverse Analysis the new airfoil profile is shown in:

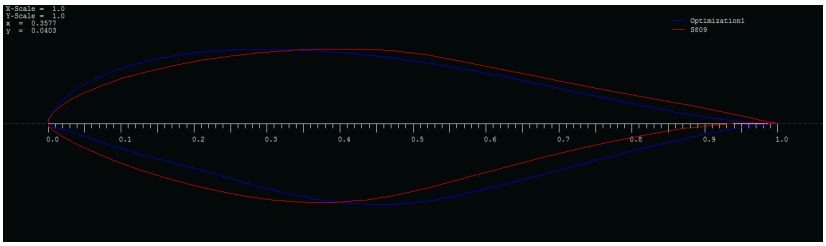


Figure 54: Scenario 1 optimized profile

Pressure distribution New airfoil @ 10^{o}

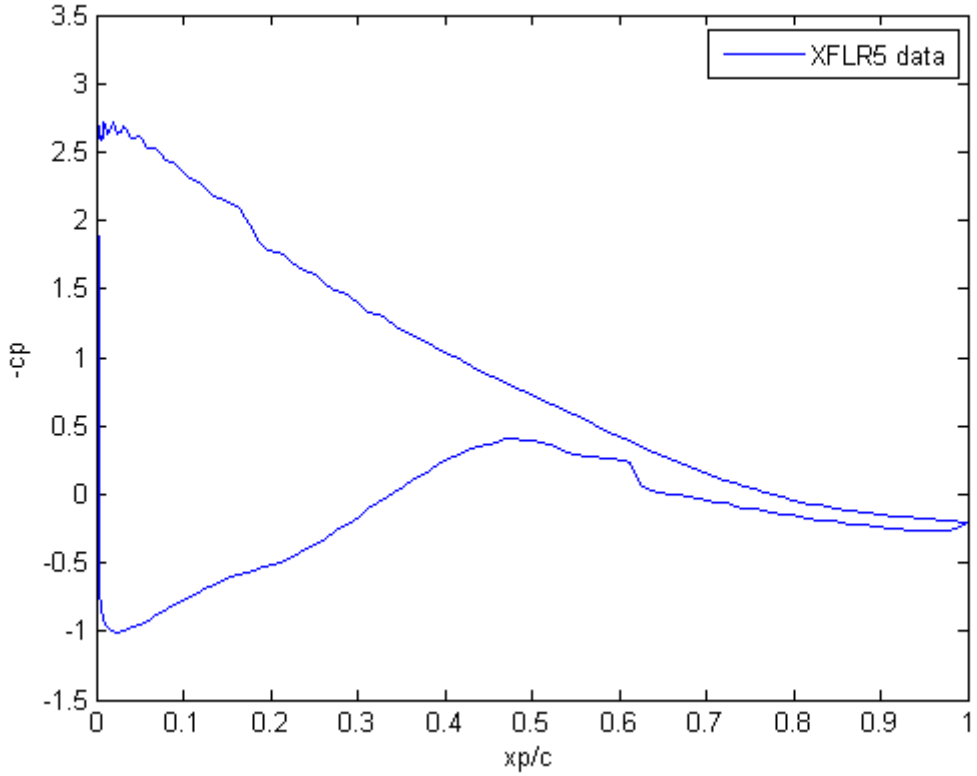


Figure 55: Scenario 1 pressure distribution

5.3.2 Scenario 2

Using Inverse Analysis New airfoil

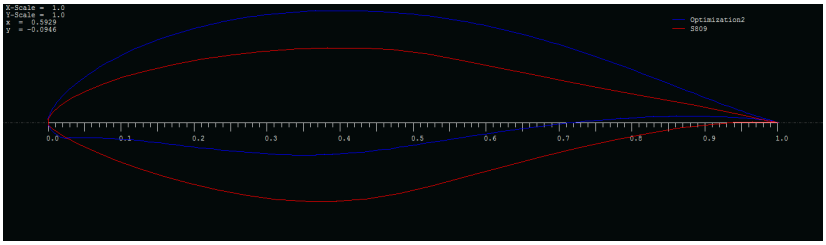


Figure 56: Scenario 2 optimized profile

Pressure distribution New airfoil @ 10^{o}

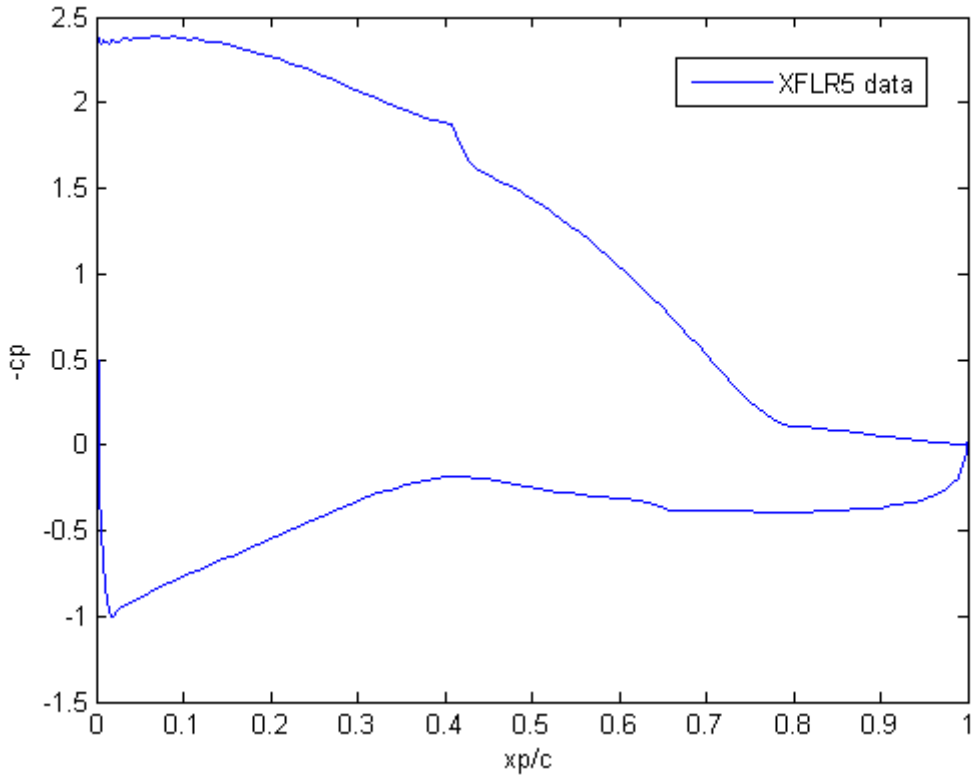


Figure 57: Scenario 2 pressure distribution

5.4 Final results

Comparison of XFLR5 vs OpenFoam Original airfoil Cp distribution and separation point estimation:

- Original S809 airfoil: it can be appreciated that the pressure peak on the leading edge is smaller because of the contribution of the viscous effects. The separation point computed with XFLR5 and OpenFoam is the same and it is located at 0.47c.

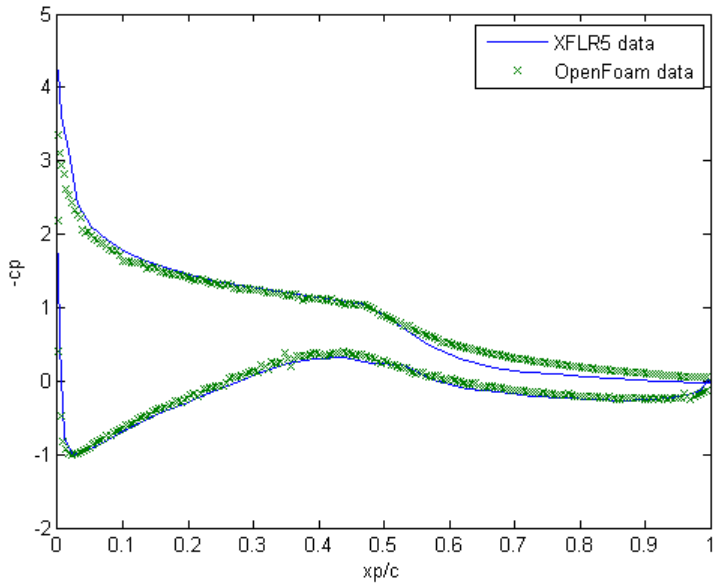


Figure 58: S809 pressure distribution comparison

- Scenario 1: for this case, the pressure drag coefficient behavior was improved and that is reflected on the fact that the separation point is delayed until 0.62c.

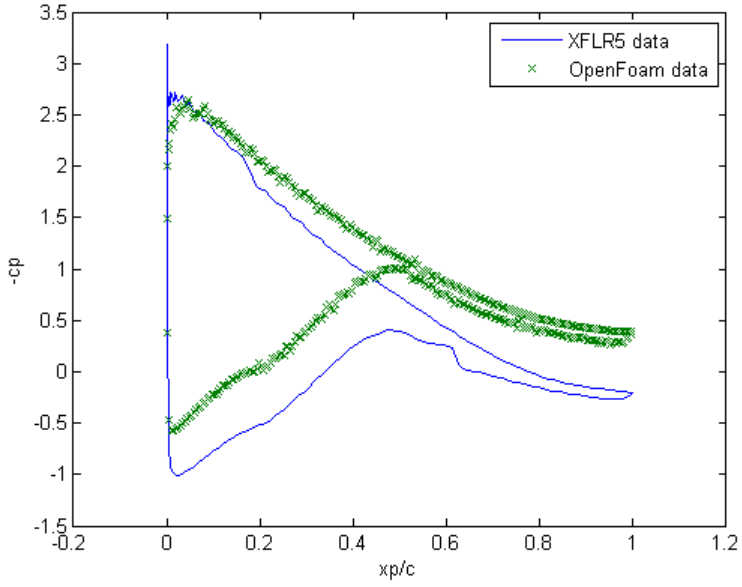


Figure 59: Scenario 1 pressure distribution comparison

- Scenario 2: in this final simulation, since the main goal was to improve the aerodynamic efficiency by increasing the camber of the airfoil, the separation point is indeed delayed up to 0.8c. This value is similar for both XFLR and OpenFoam results.

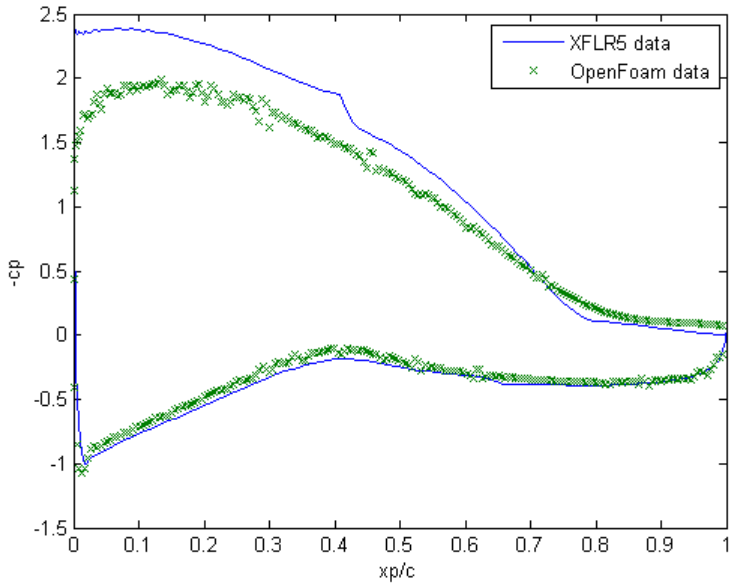


Figure 60: Scenario 2 pressure distribution comparison

The main comparisons analyzed refer to lift coefficient, polars, pressure drag coefficient and aerodynamic efficiency improvements. These are the most relevant parameters of this study and, therefore, it is important to verify that the methodology proposed is useful and provides the expected results.

1. Lift coefficient comparison of XFLR and OpenFoam results:

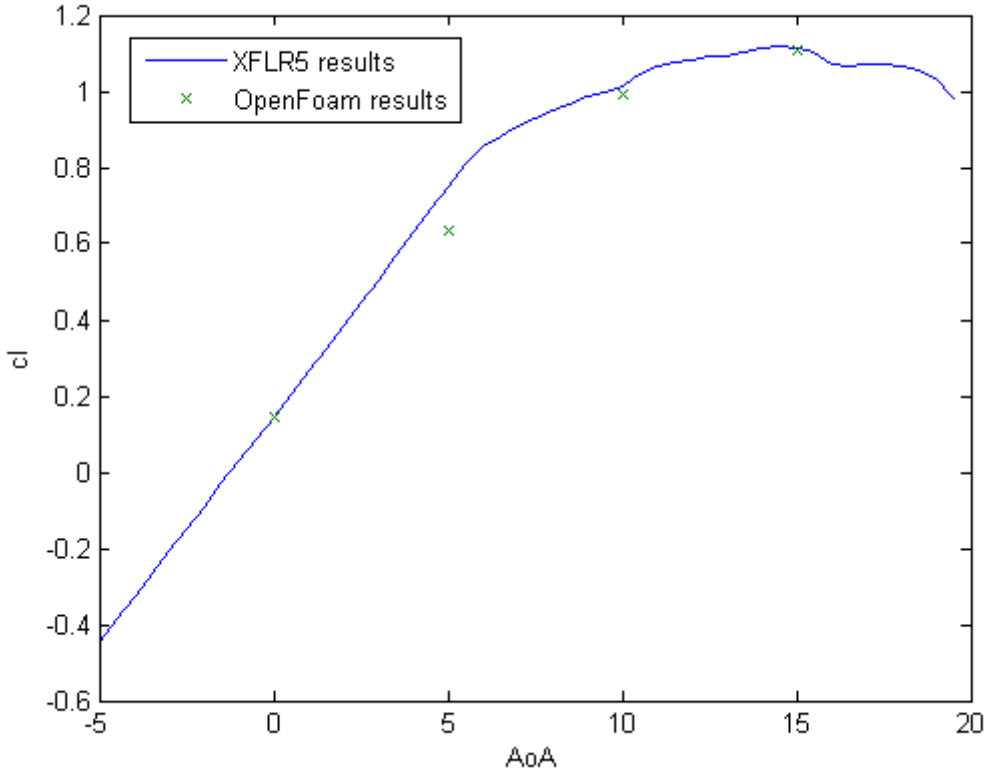


Figure 61: Lift coefficient S809

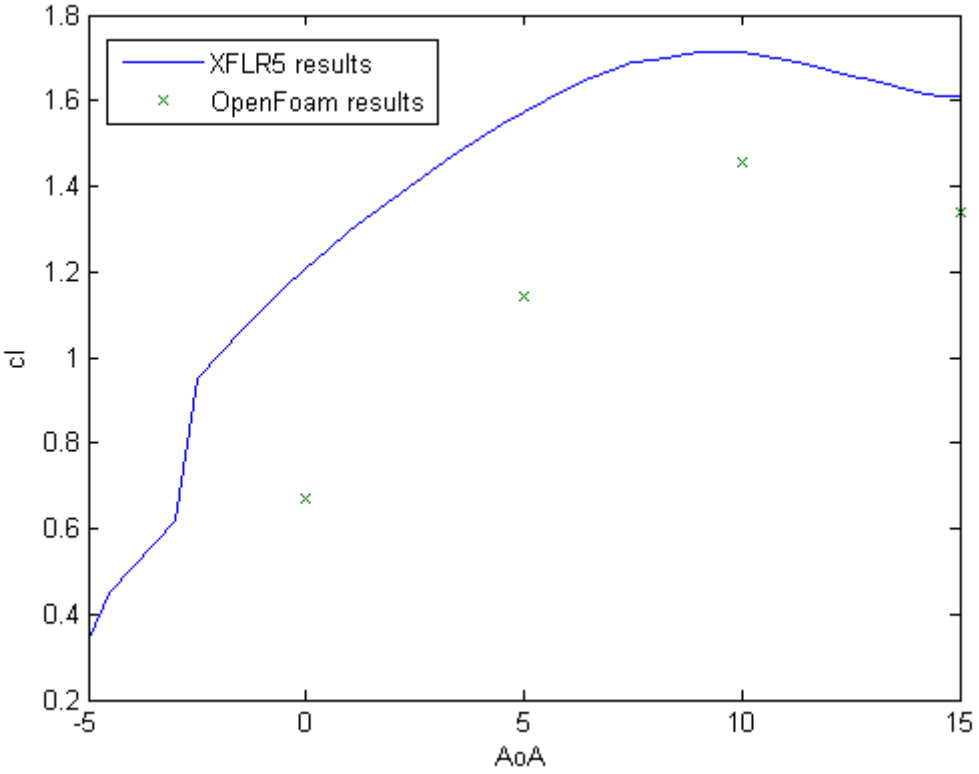


Figure 62: Lift coefficient Scenario1

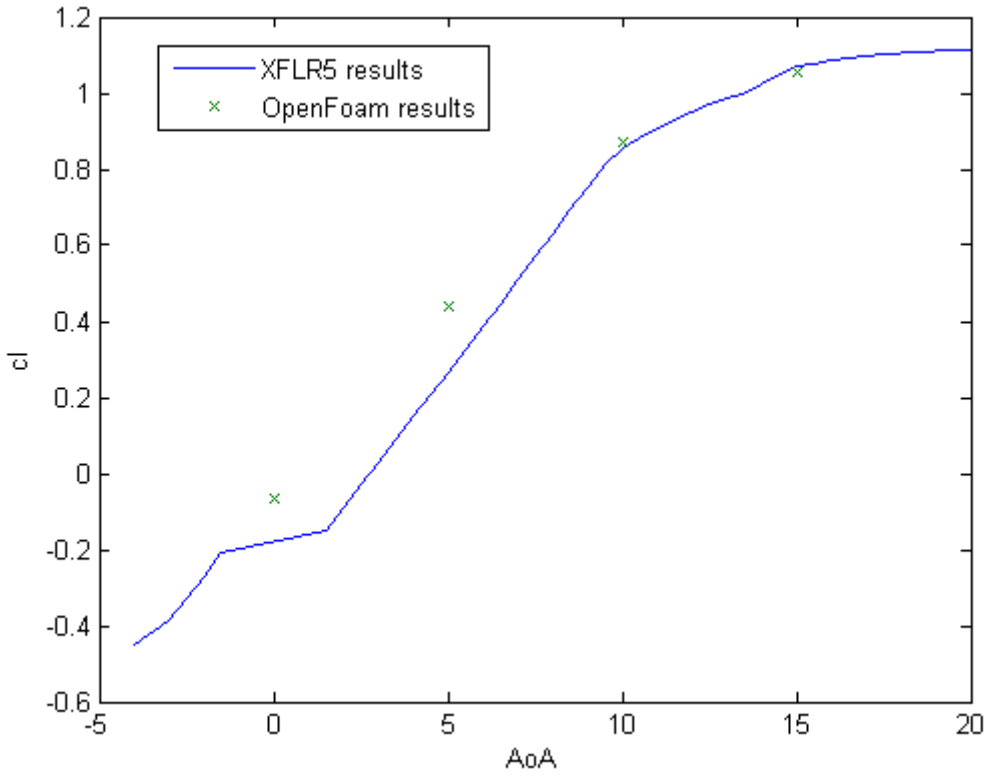


Figure 63: Lift coefficient Scenario2

The difference between the potential solution and the RANS simulation is due to the effect of the viscous terms on the Navier-Stoke's equations. Those terms generate the differences between both methods that are shown in the result. On the other hand, the data points obtained from OpenFoam fit the expected linear region of the lift coefficient and also satisfy the evolution near the stall region.

2. Pressure drag coefficient improvement:

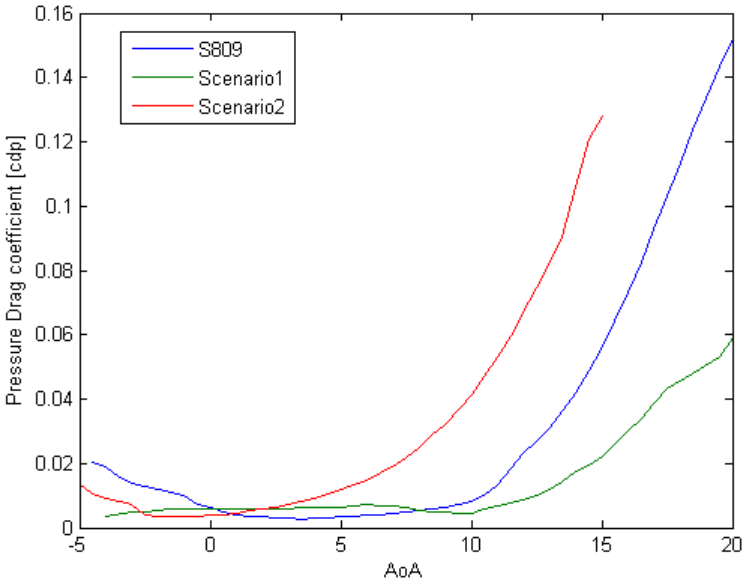


Figure 64: Pressure drag coefficient comparison

It can be appreciated that for the desired angle of attack of this project (10^o), Scenario 1 airfoil presents an improved behavior with lower pressure drag coefficient at that value, making losses smaller and improving the airfoil performance in terms of separation. On the other hand, Scenario 2 breaks the compromise, increasing camber and thickness produces higher losses due to pressure and the value of the pressure drag coefficient at 10^o is higher that the one belonging to the original airfoil.

3. Aerodynamic efficiency improvement:

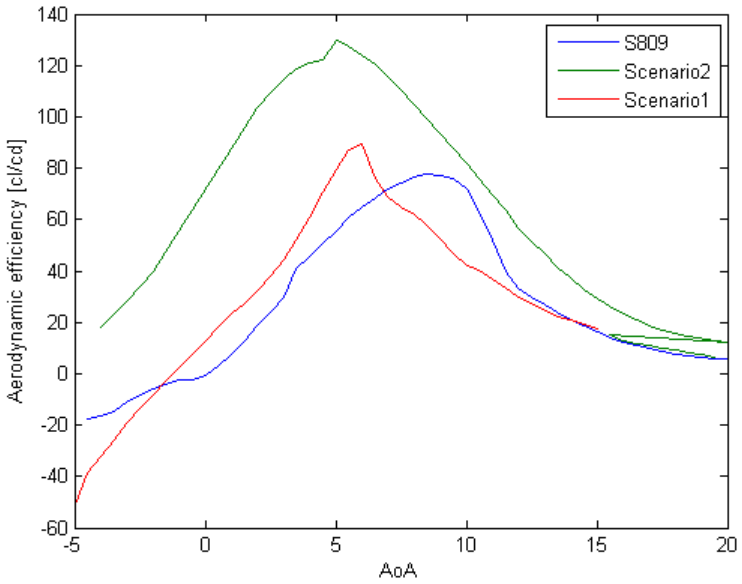


Figure 65: Aerodynamic efficiency comparison

. In this figure it is important to remark that both optimized profiles present better aerodynamic performance than the original one. The reason for that is, in the first case due to the effect of delayed boundary layer and in the second one due to the increase in camber and thickness. Scenario 2 also exhibits improved aerodynamic performance due to the fact that the stall angle has been delayed as it is shown in the . That also means that the separation point is delayed as it can be appreciated on Figure 41.

6 Conclusions

6.1 Software Validation and Advantages

One of the main goals of this project was to check whether open-source software was suitable for analysis of 2D aerodynamic profiles. It is important to remark that XFLR5 is based on potential methods whereas OpenFoam is based on RANS according to the selection for this project. That implies that the results obtained from OpenFoam, are results for turbulent analysis with a 2D flow with mean velocity and 2D turbulent fluctuations while XFLR5 results correspond to potential flow analysis. The selection of XFLR5 as software for this project, was due to the fact that potential methods offer an effective tool for profile optimization. It was shown in the results that the drag of the improved airfoil profiles was on the same order of magnitude than the original one and in the meantime, the lift of the optimized profiles has been improved substantially. For the first scenario that is due to the improvement on the profile itself, and for the second scenario it has a contribution of the increase on camber of the profile. That second improvement might seem obvious because by increasing camber, separation point and stall are delayed, but as it is shown on the c_l vs. α graphics, the lift coefficient curve is not only displaced up, but also shifted to the

right; and that is due to the fact that there has been indeed an improvement on boundary layer separation point.

The difference between the computational methods that each software is using is critical to understand the differences between the results provided from one and the other. In XFLR5, the viscous terms on the Navier-Stoke's equations are neglected. On the other hand, OpenFoam does not just consider those viscous effects, but computes the effect that viscosity has on the forces and pressure distribution around our profile.

Since this project was a first approach to simulation of 2D airfoils with OpenFoam, one of the major handicaps that arose during the development of the project was the slow learning-curve of this software. Because of that, this project is also trying to provide further users of OpenFoam with a tutorial on how to perform this type of analysis, or any other, because it faces every single step on the process to obtain the desired results. That is an important remark, because the post-processing of the results and its interpretation becomes a difficult task for early-adopters of this software.

It is important also to understand that the conditions for each one of the simulations due to the fact that each software is based on a different mathematical model explains the difference that can be appreciated on pressure distribution graphs and lift coefficient results.

6.2 Improvement on boundary layer separation

The relevant contribution of this project is that it provides students from different levels interested on running this type of projects with open-source tools instead of using commercial software and obtain reliable results from the simulations.

Two different lines have been followed in this project: the first one was focused on the pressure drag decrease that would result on delay of the boundary layer separation point. According to the results obtained, a 15% chord delay was obtained. For the second line, the goal was to improve the aerodynamic efficiency in order to be able to provide more power and, therefore, increase the profitability of wind turbines. For those purposes, Scenario2 results on an improved airfoil with more camber than the original one but with similar area and geometry. In this case, the separation point was delayed 33% of the chord until reaching a position really close to the trailing edge. That results in lower values of drag coefficients and in an increase on P_T due to the improvement on aerodynamic efficiency (η).

6.3 Economic impact estimation

According to Equation (3), an improvement on aerodynamic efficiency would imply an improvement in power production, leading to more electricity generation for a given turbine and, therefore, increased revenues for the turbine manufacturer company. Since one of the main costs of manufacturing is the material, throughout this project, the area of the aerodynamic profile has been kept almost constant without big variations in order to make it profitable for the manufacturers to change from one design to another without increasing raw material or machining costs.

6.4 Next steps

It is also important to remark that this has been a first approach to the use of OpenFoam. It is a software with enormous capabilities and for further projects, it will be really useful to present the basis on how a simple case should be set in order to move forward with problems with more complexity. The next reasonable step for this project would be to refine the analysis and obtain the separation point by one of the methods that are provided on [17].

Besides that, a finer analysis could be performed by improving the mesh, refining the time-integration intervals and also improving the Spallart-Allmaras default method provided by OpenFoam.

Using OpenFoam capabilities, for Master Thesis or PhD students, the whole 3D blade model with twist angle and a redefinition of the variables for finer analysis could be a project of interest. In order to make it useful for the industry, it would be also interesting to analyze the effect of a whole 3-blade system with real aerodynamic profiles, simulating a real wind turbine with rotation effects and compare the results obtained from OpenFoam with the ones obtained from corporate private software.

7 Appendix I. Software input data files

7.1 XFLR5 airfoil input data file

Obtained from [19], the input data file containing the S809 airfoil points is shown below. This data structure should be saved as a ".dat" file in order to be processed by XFLR5. As it was mentioned in Section 4.1.1, the data points start at the trailing edge and move counter-clockwise through the upper surface of the airfoil, reaching leading edge point and then moving through the lower surface of the airfoil until they reach the trailing edge point. If the data does not follow that order, the airfoil would not be processed properly and we would encounter some error when opening the file with XFLR5.

1.000000 0.000000
0.996203 0.000487
0.985190 0.002373
0.967844 0.005960
0.945073 0.011024
0.917488 0.017033
0.885293 0.023458
0.848455 0.030280
0.807470 0.037766
0.763042 0.045974
0.715952 0.054872
0.667064 0.064353
0.617331 0.074214
0.567830 0.084095
0.519832 0.093268
0.474243 0.099392
0.428461 0.101760
0.382612 0.101840
0.337260 0.100070
0.292970 0.096703
0.250247 0.091908
0.209576 0.085851
0.171409 0.078687
0.136174 0.070580
0.104263 0.061697
0.076035 0.052224
0.051823 0.042352
0.031910 0.032299
0.016590 0.022290
0.006026 0.012615
0.000658 0.003723
0.000204 0.001942
0.000000 -0.000020
0.000213 -0.001794
0.001045 -0.003477

0.001208 -0.003724
0.002398 -0.005266
0.009313 -0.011499
0.023230 -0.020399
0.042320 -0.030269
0.065877 -0.040821
0.093426 -0.051923
0.124111 -0.063082
0.157653 -0.073730
0.193738 -0.083567
0.231914 -0.092442
0.271438 -0.099905
0.311968 -0.105281
0.353370 -0.108181
0.395329 -0.108011
0.438273 -0.104552
0.481920 -0.097347
0.527928 -0.086571
0.576211 -0.073979
0.626092 -0.060644
0.676744 -0.047441
0.727211 -0.035100
0.776432 -0.024204
0.823285 -0.015163
0.866630 -0.008204
0.905365 -0.003363
0.938474 -0.000487
0.965086 0.000743
0.984478 0.000775
0.996141 0.000290
1.000000 0.000000

7.2 OpenFoam airfoil input data file

The case for OpenFoam input data is more complex. In Section 4.2.1, all the different set-up parameters for the required files were explained. However, there is a required file for OpenFoam to be able to run the case: the mesh file. That file comprises all the nodes, lines and elements that all together conform the mesh that was shown in Section 3.1. Mesh file has huge dimensions according to the huge number of elements that are being analyzed, that is the reason why that file will not be fully attached to this document but, in the event the reader is interested in seeing those files, all the mesh files that were employed for this project will be available for review.

8 Appendix II: OpenFoam Tutorial

8.1 What is OpenFoam?

OpenFoam is a C++ library, used to create applications that can be divided into two different groups: solvers which contain the solution to the equations used to solve continuum mechanics problems, and utilities which are used for data manipulation activities (pre- and post-processing). OpenFoam is developed to run only for Lynux Operating Systems (OS). If the computer used for calculations with OpenFoam does not run a Linux machine, such us Ubuntu, the analysis cannot be performed. Users that do not run Linux OS can download Ubuntu operating system at the official Ubuntu website [21]. Once download is complete, see the installation guide [22].

After finishing this process, user is allowed to install OpenFoam by following the steps shown in Section 8.2.

8.2 How to install OpenFoam on your computer?

In order to install OpenFoam and the post-processor ParaView, the following steps are required:

1. Go to OpenFoam official website [23] to download the desired software.
2. Select the desired version of the software you want to download. For the purposes of this project Ubuntu Deb Pack Version 2.3.0 was used.
3. User must be log in with administrator permissions in Ubuntu once it is installed in order to install the executable files with the command "sudo" that will install OpenFoam and ParaView.
4. Open Terminal and type the instructions provided in the OpenFoam official website: [24].

```
VERS=$(lsb_ release -cs)
sudo sh -c "echo deb http://www.openfoam.org/download/ubuntu $VERS main >
/etc/apt/sources.list.d/openfoam.list"
sudo apt-get update
sudo apt-get install openfoam230 (This line installs OpenFOAM Version 2.3.0)
sudo apt-get install paraviewopenfoam410
```

After typing all these commands, OpenFOAM 2.3.0 along with ParaView Version 4.1.0 are installed.

Note: If any issue arises while installation process, refer to OpenFoam official website [7] and documentation [24] where solutions to most of the issues users confront might be found.

8.3 How to run a case using OpenFoam?

In order to run a case using OpenFoam, the following steps are required (the example provided is the one used to obtain the solution of this project):

1. Understand the physical phenomena that is being studied.
2. Select the appropriate solver for that case. Further explanation on the detailed solver that was suitable for this project is provided in Section 8.4.
3. Create a directory where the case is going to be run.
4. Add the required files (Initial conditions, Boundaries, Mesh and Solver) for the case to be run. This is probably the most critical part, a dedicated section (Section 8.3) focuses on this step for the simulation performed to solve this project.
5. Run the case.
6. Post-processing of the results.

8.4 Solver selection

OpenFoam offers a long list of standard solvers created by the development community. For the purposes of this project, the solvers of interest are those regarding Incompressible Flows. According to the official website [25], a reduced set of solvers that could be used in order to solve this analysis is listed below:

adjointShapeOptimizationFoam: Steady-state solver for incompressible, turbulent flow of non-Newtonian fluids with optimization of duct shape by applying "blockage" in regions causing pressure loss

boundaryFoam: Steady-state solver for incompressible, 1D turbulent flow, typically to generate boundary layer conditions at an inlet, for use in a simulation

channelFoam: Incompressible LES solver for flow in a channel

icoFoam: Transient solver for incompressible, laminar flow of Newtonian fluids

MRFSimpleFoam: Steady-state solver for incompressible, turbulent flow of non-Newtonian fluids with MRF regions

pisoFoam: Transient solver for incompressible flow

simpleFoam: Steady-state solver for incompressible, turbulent flow

SRFSimpleFoam: Steady-state solver for incompressible, turbulent flow of non-Newtonian fluids in a single rotating frame

windSimpleFoam: Steady-state solver for incompressible, turbulent flow with external source in the momentum equation

According to the purposes of this project, "simpleFoam" solver was chosen as solver for the flow over the S809 airfoil profile. As it was mentioned on Sections 6.1 and 6.4, for further studies and improvement of the solution with the full 3D model of the turbine blade, SRFSimpleFoam could be used to model the rotation of the whole turbine. In order to solve the problem, user should only type on the Terminal window of Ubuntu, the desired command as it is shown on Figure 66.

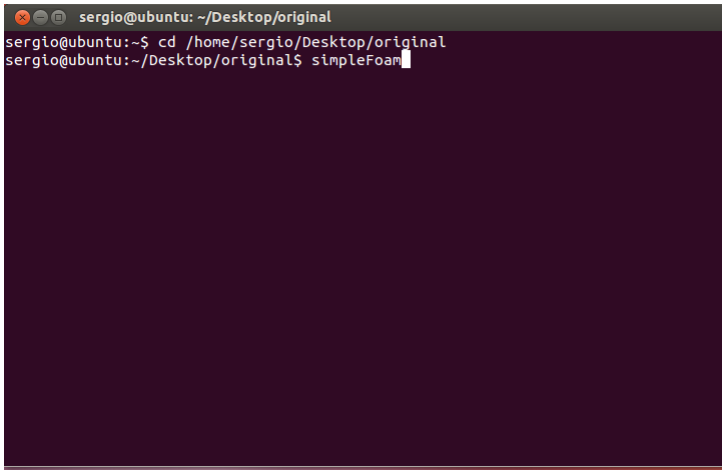


Figure 66: Solver execution

After all the data files have been prepared and contain the desired parameters for the simulation to be run. It needs to be mentioned that each solver requires its own data file set-up in order to run properly. The adequate data files for simpleFoam solver will be covered in the next section of this tutorial.

8.5 Files used in the set-up of an OpenFoam case

In this section, the different files required to run a case and their format, required for the proper solution of the case, are explained for the understanding of the reader. OpenFoam cases need to be contained in a singular folder named case "directory". On each case directory, OpenFoam requires

3 folders and 2 files. The folders contain the files corresponding to initial and boundary conditions, mesh definition and solver and, finally system settings. These folders are shown in Figure 67 .

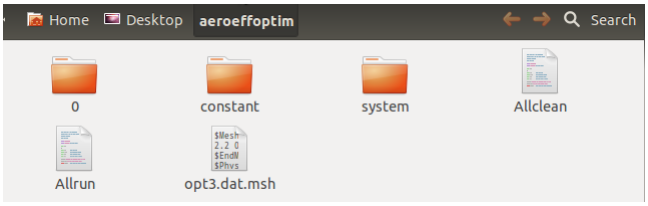


Figure 67: OpenFoam Directory

O folder contains initial and boundary conditions with the values of the velocity and pressure distributions on the different surfaces of the mesh. Each surface may have different properties and values depending on the configuration of the problem.

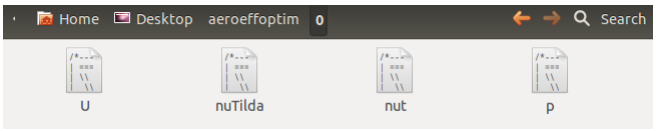


Figure 68: Initial conditions folder

System folder contains the basic files for OpenFoam cases to be run, these are system default files that users will not modify unless they want to change the internal functions of the solver.

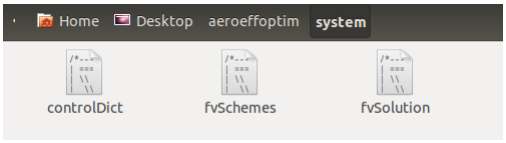


Figure 69: System folder

The directory contains also the files that are used to "clean-up" the case for subsequent simulations. They are called "AllClear" and "AllRun" and they are shown in Figure 67.

8.6 Post-processing: results' files and ParaFoam

In order to have a graphic view of the results after OpenFoam solves the case, ParaFoam application is required. This post-processing tools provides the graphic view by simply typing the command "ParaFoam" on the Ubuntu terminal. A new screen will appear and users will see something similar to Figure 70.

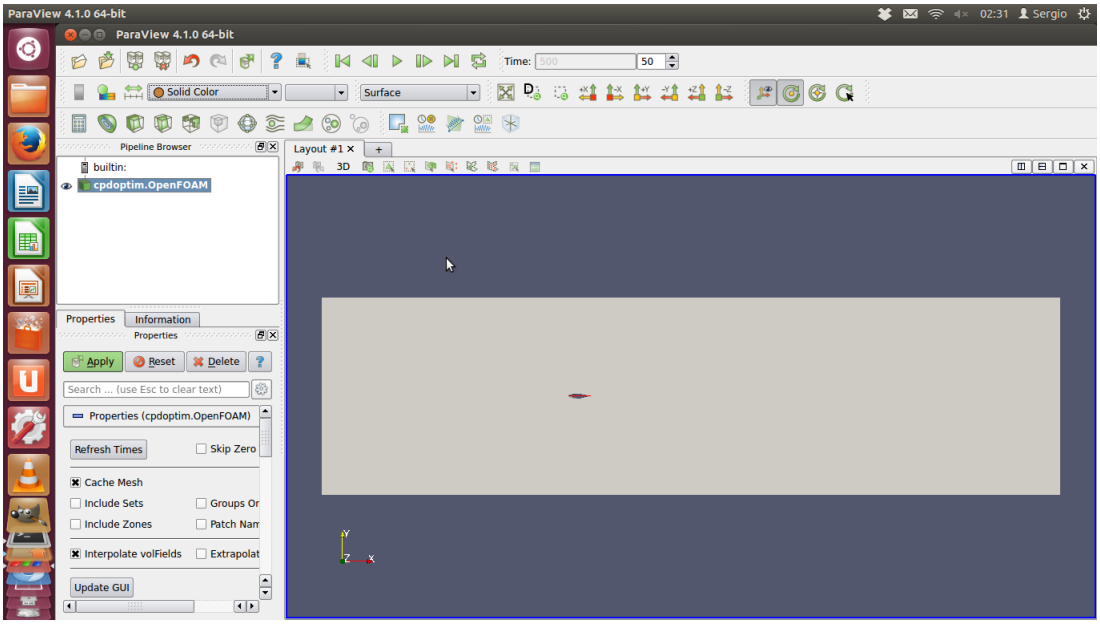


Figure 70: ParaFoam main screen

In this screen users can choose among the different solutions (mainly p and U) and the different display options. They will also be allowed to choose among different display options: contours and streamlines were the ones used for this project and they will be allowed also to modify the color scales according to their needs. There is also the option to export the data of the whole domain for further post-processing.
With these instructions, users will be able to run their own cases and visualize their results or obtain the required data for deeper analysis.

References

[1] Eize de Vries. Maintainance of turbine blades. http://www.windpowermonthly.com/article/1137943/service-maintain-wind-turbine-blade, July 2012.

[2] Yasantha Pathirana. Integral fuel tanks.http://aviamech.blogspot.com.es/2011/04/integral-fuel-tanks.html, April 2011.

[3] Electric Spanish Network. Total demand. https://demanda.ree.es/movil/peninsula/demanda/total, August 2014.

[4] M.King Hubbert. *Nuclear Energy and Fossil Fuel. Publication No. 95 Shell Development Company.* American Petroleum Institute, 1956.

[5] Energy Information Administration. http://www.eia.gov/, August 2014.

[6] ANSYS Fluent. Ansys fluent.http://www.ansys.com/products/simulation+technology/fluid+dynamics/fluid+dy

[7] OpenFoam. Openfoam software. http://www.openfoam.com/.

[8] XFLR. Xflr5 software. http://www.xflr5.com/xflr5.htm.

[9] Alex Kalmikov and Katherine Dykes. *Wind Power Fundamentals.* PhD thesis, MIT Wind Energy Group.

[10] Martin O.L. Hansen. *Aerodynamics of Wind Turbines.* EarthScan, 2008.

[11] A. Betz. *Introduction to the Theory of Flow Machines.* Oxford: Pergamon Press, 1966.

[12] Boe: Ley 24/2013, de 26 de diciembre, del sector eléctrico, 2013.

[13] Aldo V. da Rosa. *Fundamentals of Renewable Energy*, chapter Ch. 15. Elsevier Inc.

[14] Julia Layton. How wind power works: Turbine aerodynamics. http://science.howstuffworks.com/environmental/green-science/wind-power3.htm.

[15] Joshua Foss. Let's go fly a wind turbine! http://www.metrohippie.com/lets-go-fly-a-wind-turbine/, December 2007.

[16] Elemental Energy Technologies Limited. Vortex axial flow turbine. Technical report.

[17] Antonio Luis Sánchez. Chapter 10: Boundary layer. lecture notes.

[18] Dan M. Somers. Desing and experimental results for the s809 airfoil. Technical report, State College, Pennsylvania, 1997.

[19] Nrel's s809 airfoil. http://airfoiltools.com/airfoil/details?airfoil=s809-nr.

[20] Grant Ingram. The durham gmsh totorial. Technical report, Durham University, 2012.

[21] Ubuntu. Ubuntu download. http://www.ubuntu.com/download/desktop.

[22] Ubuntu. Ubuntu installation guide. https://help.ubuntu.com/community/graphicalinstall.

[23] OpenFoam. Openfoam download. http://www.openfoam.org/download/.

[24] OpenFoam. Openfoam documentation. http://www.openfoam.org/docs/.

[25] OpenFoam. Openfoam solvers. http://www.openfoam.org/features/standard-solvers.php.

[26] Boyle. *Renewable Energy*. 2004.

[27] Prof. Dr. Norbert Ebeling. Boundary layer theory.

[28] R.R. Ramsay J.M. Janniszewska & G.M. Gregorek. Wind tunnel testing of three s809 aileron configurations for use on hawts. Technical report, Ohio State University, 1996.

[29] Grant Ingram. The durham openfoam tutorial. Technical report, Durham University, 2012.

[30] Prabhanshu Pavecha. Turbulent flows and their theoretical treatments. Technical report, Indian Institite of Technology, 2012.

[31] Stephen Pope. *Turbulent Flows*. Cambridge, 2000.

[32] Dr. Hermann Schlichting. *Boundary-Layer Theory*. McGraw Hill, 1979.

[33] J.L. Tangler & D.M. Somers. Nrel airfoil families for hawts. Technical report, 1995.

[34] Acciona Windpower. Aw-3000. Technical report, Acciona Windpower.

[35] Ze Zhang. *Flow simulation over 2D airfoil using OpenFoam*. PhD thesis, Delft University of Technology, 2012.